AFIT/GOR/ENS/93M-26

THE USE OF SIMULATION TO EVALUATE STRATEGIC AEROMEDICAL EVACUATION POLICY AND PLANNING

THESIS

Charles W. Wolfe, Jr.
Major, USAF
AFIT/GOR/ENS/93M-26

Approved for public release; distribution unlimited

Thesis Approval

Student: Major Charles W. Wolfe, Jr. Class: GOR-93M

Thesis Title: The Use of Simulation to Evaluate Strategic Aeromedical Evacuation Policy and Planning

Defense Date: 26 February 1993

Committee:	Name/Department	Signature:
Advisor	Dr. Edward F. Mykytka Associate Professor Department of Operational Sciences	Edward F. Mykytka
Reader	Colonel Thomas F. Schuppe Dean School of Logistics & Acquisition Management	Thomas F Schuppe

AFIT/GOR/ENS/93M-26

THE USE OF SIMULATION
TO EVALUATE STRATEGIC AEROMEDICAL EVACUATION
POLICY AND PLANNING

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Operations Research

Charles W. Wolfe, Jr., B.S., M.B.A.
Major, USAF

March 1993

Approved for public release; distribution unlimited

Preface

This thesis develops and documents a computer simulation model that incorporates the major elements of strategic aeromedical evacuation (AE) and presents an initial analysis of simulation output for a specific scenario. This model is modular, completely data driven, and easily adaptable to evaluate differing scenarios and associated aeromedical evacuation policies and plans. The Air Mobility Command Analysis Group (AMC/XPY) can use this tool and information to assist the AMC Surgeon and his staff to improve strategic AE contingency planning and thus its eventual execution.

I would like to first thank my advisor, Dr Ed Mykytka, for his invaluable support, insights, and guidance during this process. I would also like to thank the other members of my committee, Col Tom Schuppe and Major John Borsi. Col Schuppe did a great job teaching me the SIMSCRIPT language and helped me battle through the code. John Borsi, my long-time friend, suggested the topic to me and has been a true source of encouragment throughout my AFIT experience. Although not on my committee, I would like to credit Lt Col Ken Bauer for suggesting the idea of using multivariate techniques to analyze the simulation output.

This thesis was sponsored by the AMC Analysis Group. In particular, I would like to thank Lt Col Joe Litko, who guided this effort based on his personal experiences modeling contingency airlift operations during Grenada, Desert Storm, and Somalia. The value he added was immeasurable. Special thanks also to Mr Keith Ware, former member of the

group, and Mr Alan Whisman for their assistance in helping me start this effort and introducing me to the concepts of aeromedical evacuation.

Many in the medical community also helped with this research. A note of thanks deservedly goes to Col Carroll Bloomquist and Major Phil Mahlem of the AE Medical Plans and Requirements Office at AMC. They graciously gave of their time to teach me the complexities of the AE business and provided key scenario data as well as their expertise on the subject. Thanks also to Lt Col Sam Hernandez (US Army), Lt Col Bruce Bossart, and their staff at the Armed Services Medical Regulating Office for sharing their knowledge of the medical regulating process.

My greatest thanks go to my two children, Katheryn and Matthew, for giving up their time with daddy and to my wife Geri, who once again has given her time, energy and love to help me through a significant challenge. I am particularly proud of her, because as an Air Force Reserve nurse, she temporarily gave up being a mom and wife to serve in Oman during Desert Storm. And today she, along with thousands of others, again stand ready to carry out this important mission.

Table of Contents

List of Figures

Figure Page

List of Tables

AFIT/GOR/ENS/93M-26

Abstract

Strategic aeromedical evacuation (AE) of casualties from the theater of operations to the CONUS during wartime is a complex operation that involves the integration of medical personnel and policies with airlift concepts and capabilities. Military analysts within the Air Mobility Command Analysis Group (AMC/XPY) have traditionally used deterministic linear programming techniques to estimate the number of aircraft the United States Air Force (USAF) requires for given contingency scenarios. However, this group has yet to develop a stochastic approach to validate their resource recommendations, and more importantly, to study the interrelationships between key factors comprising strategic aeromedical evacuation. As the possibility for many smaller campaigns around the world increases, USAF medical planners require a flexible, analytical tool which captures the major elements of this important mission in order to quickly evaluate differing medical airlift plans and policies.

This thesis develops, documents, and demonstrates the use of a computer simulation model for strategic AE operations that is modular in nature, completely data driven, and quickly adaptable to scenario changes, as a policy/planning aid for the AMC Surgeon and his staff. In addition, this thesis investigates the use of two statistical techniques, principal component analysis and factor analysis, for interpretation of the simulation output.

THE USE OF SIMULATION
TO EVALUATE STRATEGIC AEROMEDICAL EVACUATION
POLICY AND PLANNING

I. Introduction

This research effort develops a computer simulation to model and investigate key elements of strategic aeromedical evacuation (AE) during contingency operations. Strategic AE is used primarily to airlift casualties from the theater of operations to appropriate care facilities within the United States. This study is sponsored by the Air Mobility Command Analysis Group (AMC/XPY). This chapter provides the essential background, problem statement, research objectives, and scope of this study.

Background

Strategic aeromedical evacuation has its roots in the Vietnam War when, for the first time, the United States Air Force (USAF) airlifted casualties directly from the theater of operations (Saigon) to Andrews AFB in the continental United States (CONUS), reducing the total patient travel time by as much as three days (10:1). This new concept saved countless lives. Since then, the minimization of both the travel time from the theater of operations to the CONUS and the number of times a patient is handled during this transit to a hospital has guided nearly all basic efforts tc improve strategic AE operations.

Stimulated by these two goals, in May of 1986, Congress authorized Military Airlift Command (MAC), now Air Mobility Command (AMC), to use aircraft from the Civil Reserve Air Fleet (CRAF) to accomplish strategic AE during wartime. For the first time, dedicated aircraft were assigned to this important mission. Originally, AMC contracted with the airlines for 85 Boeing 767 airframes configured for AE. Recently, the AMC analysis group has performed several resource requirement analyses which have resulted in a decision to reduce the overall number of airframes to approximately 45. These analyses were deterministic in nature, and the stochastic (or random) elements of the AE system have not been addressed (37).

It is expected that AE will play an even more visible and prominent role in future warfare. Fortunately, during the recent Gulf War, with our airlift capabilities stretched beyond their limits, our forces experienced miraculously low casualty rates. Thankfully, the question of how well the AE system could have serviced mass casualties, originally anticipated to reach into the thousands, did not demand a real answer. The next war may not prove as kind.

History. The history of aeromedical evacuation is closely tied to advancements found in the areas of medical and aviation technology. Aviation has its origins with balloon flight. As quickly as someone had devised a military purpose for the balloon, they also realized its effectiveness in transporting wounded. Thus, aeromedical evacuation was practiced as early as 1870 when, during the siege of Paris, casualties were transported by balloon to safe havens (17:392). Again, in 1910, shortly after the Wright brothers flew at Kittyhawk, Captain George Gosman of Ft Barrancas, Florida, discovered that an airplane could be used to transport the wounded (21:8). The first military medical evacuation by an aircraft was flown in April of 1918. A French medical officer named Dr Chaissang designed a modification to one of his country's military aircraft. The modification provided adequate room for two casualties located right behind the cockpit. The patients were inserted through holes in the sides of the fuselage. It performed the mission, albeit a bit chilly for the patient. Aircraft were used for this purpose only to a limited extent during World War I. The lack of practical airplanes and the relative safety of travel by air versus other means in those early years of flight most likely attributed to this (12:2-3).

The train was the primary workhorse for transportation of patients over the course of World War II (WWII), however, aeromedical evacuation began to gain widespread popularity in the latter part of the war (21:9). Table 1.1 shows how the use of AE increased toward the end of WWII.

Table 1.1. World War II Patient Evacuees by Air (35:349)

Year	Air Evacuees	% of Total Evacuees
1943	3,260	4.5
1944	31,490	18.2
1945	86,755	22.2

The transition from trains to aircraft was stimulated by a key event in January 1944. Because the local railways surrounding Stark General Hospital in South Carolina were clogged, a total of 661 patients, in 29 different plane loads, were airlifted to five neighboring general hospitals (18:56). This event unveiled the advantages of airlifting casualties and contributed to the following Departme:.t of Defense Policy in 1949:

> ... In both peace and war, the transport of patients of the Armed Forces shall be accomplished by aircraft when air transportation is available and conditions are suitable for evacuation unless medically contraindicated ... (9)

While the substantive use of aircraft to transport patients had its beginnings in WWII, it was the helicopter which made significant contributions to AE by carrying out theater tactical evacuation of patients during the Korean and Vietnam wars. However, in both wars, military cargo aircraft that could be temporarily reconfigured bore most of the workload of transporting the battle stricken. During the Korean War intratheater evacuation was accomplished primarily with the C-46, C-47, and C-54 aircraft. Theater movement could occur in one of three areas: within Korea, Korea to Japan, or within Japan (8:38). Normally, if a patient needed more than thirty days to recover, he was flown to Japan. Those going to Japan were directed to hospitals based on the type of injury they had received. The Division Surgeon tried to accomplish this match (which would eventually be known as "regulating") at the forward air strips "to permit more direct transportation and reduce the enroute time taken" (8:39). This was done on a daily basis and required close coordination between the Division Surgeon and the Military Air Transport Service (MATS). Critical information key to successfully accomplishing these missions included knowledge of battlefield events and conditions as well as the number and status of casualties already in the forward field hospitals (8:39). Intertheater evacuation took place in the form of island hopping with the C-97, from Japan through Guam, to Midway or Wake, to Hickam AFB in Hawaii, and finally to Travis AFB in California.

The C-7, C-118, C-123, and C-130 aircraft flew intratheater operations in Vietnam. It is interesting to note that more than 65 percent of all the aeromedical evacuation missions within Vietnam were unscheduled (3:281). Approximately eleven times each day, a tactical medivac mission was flown. Each of these missions consisted of coordinating requirements, identifying a medical crew, reconfiguring the cargo plane, and the flying the mission itself. Naturally, this demanded an immense amount of coordination between the aeromedical evacuation centers (AECCs), the airlifters, and the medical facilities (29:21-25).

As mentioned earlier, Vietnam provided the first opportunity for a nonstop theater to CONUS flight. While this was good for some patients, MAC soon learned that this long trip was just too demanding for others (12:24). Medical technology inside the aircraft was not quite able to provide an adequate environment for such a duration. Since the Vietnam era, there has been an extensive effort to bring more medical technology and comfort inside the aircraft in order to support longer flights. Chapter 2 describes some of these technologies.

To gain an appreciation for the level of demand placed on AE during the Vietnam War, one only need to study the operations associated with the Tet Offensive when, in the first six months of 1968, approximately 55,000 patients were moved out of country.

Then, in the 1969 Spring Offensive, nearly 11,000 patients a month were evacuated. This represents the highest demand ever placed on U.S. AE operations (12:24-25).

Perhaps the best way to capture what AE meant to commanders during this period of time is to examine some of their comments. General M.S. White, an early advocate of AE, identified the most important benefit of AE as the morale boost that it provided the fighting man (40). Concerning Vietnam operations, Lt General Kenneth Pletcher (the USAF Surgeon General) said, "thousands of U.S. fighting men are alive today because speed, new techniques, and trained personnel of aeromedical evacuation teams gave the wounded in Vietnam better than twice the chance of survival than ever before" (29:17).

AE operations during Korea and Vietnam provided two primary lessons. First, the operations highlighted a need for dedicated intertheater aircraft. The second lesson was the realization that the most effective use of AE resources came when under the control of a single command (8:121-124).

While aeromedical evacuation has a proud history and many accomplishments to its credit, the future holds the potential for even greater requirements and challenges. The massive firepower and aggressive tactics associated with today's weaponry hold the potential to deliver a much greater magnitude of human catastrophe in a much shorter period of time than has ever been experienced before.

Concepts of Aeromedical Evacuation. The aeromedical evacuation mission is the responsibility of Air Mobility Command. Wartime AE can be defined as the medically supervised movement of casualties by air transportation to and between medical treatment facilities (MTFs) (11:3). AE seeks to improve casualty recovery rates and sustain the morale of combat forces by providing those forces the knowledge that lifesaving medical resources are available and can be quickly and effectively provided to any location in the world (11:3).

The Air Force's AE operations are conducted in three major areas: intertheater, intratheater, and domestic. Intratheater AE, also known as tactical AE, transports patients (primarily using C-130 aircraft) between MTFs located within the combat zone (area needed by combat forces to conduct operations) in the theater of operations. Intertheater, or strategic AE, is the transport of casualties from an APOE in the theater of operations to the CONUS. Domestic, or CONUS redistribution of patients (using C-9 aircraft) to their final destinations is the third type of AE. This study focuses on strategic AE operations carried out by the Boeing 767 aircraft.

Management of casualties from the theater to the CONUS is accomplished through a multi-echelon system of care. The five separate echelons are distinguished by the level of care that each echelon is capable of providing. The first echelon (1E) resides on the battlefield at the point of contact and is characterized by self aid or buddy care (15:4). The second echelon (2E) provides emergency treatment and tries to return minimally injured casualties to duty as soon as possible. Those who can't be returned to duty are stabilized for movement to a higher echelon facility (15:19). Movement from 2E facilities to third echelon (3E) facilities is normally the responsibility of the parent service (11:7). The purpose of a 3E facility is to provide surgical and other specialty care within the combat zone. Fourth echelon facilities, located within the communications zone (rear part of the theater of operations), offer complete medical facilities including enhanced surgical and other medical subspecialties (15:4). Finally, hospitals located within the CONUS represent the fifth echelon (5E). Transportation from 4E to 5E facilities will be carried out by retrograde (reconfigured for medical use) C-141/C-17 or dedicated Boeing 767 aircraft from the CRAF. This study focuses on operations at the third echelon and higher.

CONUS hospitals consist of DOD, Veterans Administration (VA), and civilian hospitals within the National Disaster Medical System (NDMS) (15:4). The NDMS is a national plan to care for the victims of large-scale natural disasters. For instance, the plan calls for joint use of military and civilian resources and assumes AMC assets could be used to evacuate up to 100,000 victims of a California earthquake to cities where appropriate care could be administered. NDMS is the follow-on to the Civilian Military Contingency Hospital System (CMCHS) and will provide beds to wartime casualties (21:173).

With this basic framework in mind it is important to understand that modern strategic AE is actually nothing more than a plan, based on general policies, to employ during periods of conflict a set of resources that are used in different ways during peacetime. To help illustrate, consider the primary aircraft for strategic aeromedical airlift, the Boeing 767. These aircraft are presently airliners that will come from the CRAF. Likewise, active duty personnel make up only 7 percent of the AE forces that will execute the plan, while 93 percent will come from the Air Reserve Component (ARC) (10:5). Thus, the AE "system" doesn't presently exist for observation or experimentation. Therefore, it is critical now for AF medical planners to somehow identify and experiment with the key parameters under their control, to ensure the system will accomplish its mission in the future. Chapter 2 looks at some of the techniques analysts have built and used to gain quantitative understanding and insight into AE operations.

The biggest lesson learned from the past and from peacetime operations is the need for a single integrated manager. In a paper to the Chief of Staff of the Air Force, Air Mobility Command Surgeon (HQ AMC/SG) points out:

> A system that has single integrated management with standardized doctrine, policy, equipage, and training can best be used to transport a patient through an integrated theater and strategic system, to definitive health care facilities. Fractionization of the world wide AE system into theater parts is not consistent with sound fiscal management and threatens the precepts of centralization that allows for the maintenance of standardized doctrine, policy, training, and equipage. (10:6)

The Chief recently weighed this long standing concept of operations versus an alternative plan which gives control of tactical level AE resources to the theater commanders. The decision was made that AMC will release control of the tactical or intratheater level medical resources to the theater commander during wartime. However, centralized control of intertheater or strategic assets such as the CRAF and their crews will remain under the control of the AMC mission support structure (25). Decentralization of strategic resources promises to change and confound a set of simplifying assumptions that analysts have made when modeling the command and control of AE operations. The simulation developed in this research considered the effects of decentralized use of strategic aeromedical aircraft.

The heartbeat of aeromedical evacuation is a process known as patient regulation. Patient regulation is a selection process which matches a casualty to a hospital capable of providing the appropriate level of medical care. Regulation results in a requirement to move a specific patient to a specific hospital, as selected by the regulating office (11:5). Overall responsibility for regulation belongs to the Armed Services Medical Regulating Office (ASMRO) located at Scott AFB, Illinois. The responsibility for the care of casualties within a specific theater falls to the theater commander, who normally establishes a Joint Medical Regulation Office (JMRO) to accomplish this task. The JMRO identifies and tracks stabilized patients within the theater, finds them destination hospitals in the CONUS with the assistance of the ASMRO, and coordinates strategic AE through the ASMRO and AMC (11:6). Regulation normally occurs as a batch request from the JMRO to the ASMRO. The ASMRO identifies the needed beds in the CONUS and passes this information back to the JMRO who then coordinates with airlift for the needed transportation. During wartime regulation occurs for eight basic patient categories: medical, surgery, psychiatric, orthopedic, burns, spinal, OB/GYN, and pediatrics (5). Patients who are not regulated normally will not be placed into the AE system.

Information technology promises to change the way patient regulation occurs. As the regulating office receives and processes casualty information more quickly, the AE system will realize greater efficiencies in scheduling and routing airlift. Shared databases containing the latest patient status and instant satellite transmission of this data will provide decision makers with real-time status of casualties and their location. This will assist theater commanders with the subjective AE judgements they must make during a campaign.

One such judgement, the theater evacuation policy established by the theater commander with the advice of the theater surgeon, specifies the maximum number of days a casualty may receive treatment at facilities within the theater of operations before transfer to a CONUS hospital (11:5). The time period starts with the date of admission to the first hospital. This subjective decision helps to define AE requirements. It is a function of the number of beds available in the theater matched against the estimated number of casualties.

Problem Statement

To date, the AMC analysis group has used deterministic linear programming techniques to estimate the number of aircraft the Air Force needs for the strategic airlift of casualties during wartime. However, because of limited resources and time, the group has been unable to incorporate stochastic elements into their analyses in order to better understand the relationships between lower level parameters associated with the problem. Consequently, AMC requires a stochastic tool and an initial analysis that investigates and provides insight into what these factors are and how they influence the AE system.

Research Objectives

The purpose of this thesis is twofold. The first objective is to build and document a computer simulation model that incorporates the major elements of strategic aeromedical evacuation. Because of its expected use as a policy/planning aid, the model is required to be modular in nature, completely driven by the data, and easily adaptable to scenario changes. The model should have the "hooks" (28:1) that enable AMC/XPY to expand the simulation and attach representations of the tactical AE in theater as well as redistribution of patients in the CONUS. The second research objective is to exercise the model against an AMC/SG scenario and provide both a classical output analysis and a multivariate analysis that seeks to uncover important relationships found amongst key factors affecting strategic AE operations. In the future, the AMC analysis group can use this tool and information to assist the AMC Surgeon to improve strategic AE contingency planning and thus its eventual execution.

Scope

The scope of this research includes only strategic AE operations using the Boeing 767 from the CRAF. That is, the aircraft operations and patient movement from designated aerial port of embarkation (APOE) locations in the theater of operations to the CONUS receiving hubs. The methodology is built around the assumption that strategic AE missions are primarily demand driven, responding directly to the number of casualties requiring airlift. However, the methodology is not anticipatory and this limitation is addressed in Chapter 5. The study does not consider the tactical movement of patients in the theater or redistribution of patients in the CONUS.

This effort assumes ample maintenance support personnel, flight crews, support equipment, etc., to sustain 767 operations and to handle casualties. The study assumes the validity of its primary inputs provided by AMC/XPY, as well as the expert opinion of USAF medical planners.

II. Literature Review

This chapter highlights and summarizes some of the mathematical techniques the analytical community has exercised to help decision makers understand and evaluate the overall effectiveness of aeromedical evacuation. The chapter is divided into three main sections. The first two sections describe the two general analytic approaches, deterministic and stochastic, that have been taken to study AE. Deterministic methods are often used for evaluating peacetime elements of AE, while stochastic approaches are often used to study contingency operations. The final section of this chapter provides a sampling of the emphasis of the majority of research being conducted in the field of aeromedical evacuation. This research focuses on the improvement of highly technical, lifesaving aeromedical equipment that operates inside the aircraft.

This review primarily addresses the topic of aeromedical evacuation at the macro level, avoiding research that seeks to optimize a particular subset of the AE system. For example, a study that identifies the best internal configuration of in-flight equipment for a medivac aircraft would not contribute to this review.

Deterministic Approaches

Deterministic methods are most often used for resource sizing, route structuring, and scheduling of AE operations. Several examples of this type of research follow.

Burnes, in his thesis, *Application of Vehicle Routing Heuristics to an Aeromedical Airlift Problem,* (6) constructed a network of flight routes for an AE system, limited to thirty MD-80 aircraft operating completely within the CONUS. This research focused its attention on optimizing the redistribution of patients after their arrival in CONUS. It allocated the thirty aircraft across nine CONUS hub locations, sought optimal routes between the hubs, and monitored bed availability by type of casualty. The Clark-Wright heuristic was modified and combined with a split delivery heuristic to obtain a solution. Burnes concluded that thirty aircraft were sufficient to operate the CONUS AE system. However, he also found the flight routes generated by the heuristic were too sensitive to slight changes in patient demand and, therefore, were not suitable for an operations plan. (6:6).

In a similar effort, Carter performed a study to develop and evaluate operations plans for the MD-80 aircraft. His thesis, *Allocation and Routing of CRAF MD-80 Aircraft*, (7) used a proven probabilistic traveling salesman formulation to determine worst case routes. He then exercised the constrained number of aircraft against these routes, and concluded that thirty aircraft were sufficient for the planned operations. His results compared favorably with Burnes' implementation of the Clark-Wright algorithm (7:8-9). Again, Carter's work, like Burnes', concentrated on the adequate number and efficient routing of aircraft within CONUS.

Effort has also been focused on the scheduling aspect of AE. Whetstone, in his thesis, *A Heuristic Approach for Aeromedical Evacuation Systems Scheduling and Routing*, (39) tackled the weekly scheduling problem for peacetime CONUS AE operations. He developed a model which could be used to develop a weekly schedule but discovered that the continuously changing demand for transporting patients made it impossible to construct a schedule that was optimal for each day of the scheduling period.

He also developed a methodology to address the daily routing problem. His final scheduling heuristic produced an improved schedule. He suggested further research to investigate the effect of schedule on demand. One of his more interesting insights was that, once a fixed schedule is in place for awhile, the schedule may begin to dictate demand rather than vice versa, making existing schedules appear better than they inherently are (39:74). In his conclusions he states that "...the importance of a fixed weekly schedule should be lessened. At most there should be a flexible weekly schedule... capable of changes due to patient demands or user requirements..." (39:75-76).

Stochastic Approaches

Just as warfare, AE operations are driven by and contain many random events. The quality and flexibility of the AE policies and plans in place, as well as the people executing them, will determine the effectiveness of the system. For primarily these reasons, analysts have used stochastic approaches to provide insights into the interrelationships that exist between the major elements of AE. It is interesting to note that some of the studies described in this section either confirm or helped to establish significant AE policy.

The first study, entitled *Wartime Strategic/Domestic Aeromedical Evacuation and Distribution of Patients*, (23) was a collective effort by a research group at the Industrial College of the Armed Forces in 1982. The group, made up of students with medical related specialties or analytic skills, examined the typical scenario for that time, a NATO/Warsaw Pact conventional confrontation. Parameters of the study included a 15 day theater evacuation policy, daily arrival of 3000-5000 patients to the CONUS, and the

DOD/VA bed system in the CONUS (23:1). While details of their methodology are sketchy (a simulation model was built and exercised), their conclusions were pointed. The group concluded that retrograde (reconfiguring the C-141 in the field during operations) AE would definitely disrupt the C-141 forward deployment schedule and that, given retrograde of a C-141, its mission would then be affected by higher priority cargo movements. Summarizing, they identified an overall lack of a general strategic airlift capability. The group also said there was a need to reevaluate the principle of moving patients only as "far to the rear as the tactical or strategic situation may dictate and the patient's physical condition may permit" (23:8). Finally, they made a recommendation to further study methods to better distribute patients within the CONUS (23:9).

Many of their recommendations eventually came about. Four years later a dedicated strategic AE aircraft was obtained through means of the CRAF, and the research that followed into the redistribution of patients within the CONUS has been cited in the previous section of this chapter. Their recommendation regarding "principle of movement" resembles the approach taken in the Korean War, and since it did not support the acquisition of additional resources or technology, it probably fell on deaf ears. A recommendation to expand bed availability in the United States to include civilian hospitals was already being implemented.

A broad study, based on the wartime CONUS casualty distribution system, was accomplished by Alfano and O'Neill (1). The study specifically addressed supplementing the present C-9 fleet with planes from the CRAF. The simulation model assumed a European scenario and represented the hub-and-spoke-type distribution of patients found in the CONUS. They built a computer simulation model using the SLAM programming language and then performed a designed experiment to gain insight to the factors important to minimizing time in system for the patient (1:6).

Alfano and O'Neill identified the following as key factors: the number of patients arriving, their mean interarrival time, the number of CRAF aircraft as well as their capacity, and the number of C-9s available. They also performed a sensitivity analysis to gain further understanding of the factors over which MAC had control (1:63). Their results indicated that given a casualty arrival rate of approximately 1000 patients a day from Europe, MAC required either a 50% increase (from 11 aircraft) in the number of C-9s or four additional CRAF aircraft, each with a capacity of 175 patients (1:75).

Again, concentrating on a European scenario, Ewing, in his thesis, *Casualty Evacuation & Distribution Using B-767 and C-9 Aircraft*, (E1) built a SLAM simulation model to measure the mean time in system for a typical patient. In addition, he developed a set of response surface equations from the experimental results in order to measure the performance of the system without the need to commit additional time and money to further computer runs (16:7).

Finally, there are high resolution medical models such as the one being built by Booze, Allen, & Hamilton for the Joint Logistics Directorate (J4) at the Pentagon (4). The tool, LPXMED, simulates all of the medical processes that occur within a theater of operations (4:4). This tool will allow medical logistics planners to work in concert with operational planners to assess delivery and use of critical medical resources (4:1).

Scientific analysts continually seek innovative ways to improve aeromedical evacuation war plans utilizing a variety of techniques designed to minimize the amount of time and the number of stops a patient makes en route to an appropriate care facility in the United States. Most of these methods attempt to find the best use of a fixed amount of a resource, such as aircraft, while others try to determine the quantity of a resource required to achieve a performance objective. Others, take a broader, more probabilistic view, seeking to discover and gain insight to important relationships between key elements of the system. This type of analysis is often more flexible and able to provide leadership options and understanding in an ever changing environment.

AE Developments at the Micro Level

As previously mentioned, a great deal of the current AE research is directed toward bringing the latest medical technology to the wounded quicker than ever before. Efforts to accomplish this are primarily directed at continuing to upgrade the medical equipment inside the aeromedical airlifter. These new developments may eventually reduce the stabilization time required before a patient is declared ready for transport. This decreases total time in system for a patient but also compresses and further strains strategic AE airlift. The following are a few examples of such advancements.

Many patients require intravenous fluids and medications during flight. The Air Force has acquired a new infusion pump that generates precise fluid delivery required with the latest medications (33). Another key piece of equipment the USAF is upgrading is the pulse oximeter. This enables flight nurses to continually monitor oxygen levels in the blood, an expected standard of care (34). The University of Alabama has developed a prototype oxygen delivery system which protects the patient from hypoxemia during transport, while simultaneously conserving oxygen (30). Other promising areas lie in the development of a two-wheel gurney to allow movement of a patient by a single individual rather than four (38:6), and a new standard NATO litter made of lighter, stronger materials (27:1). These are just a few of the many projects aimed at improving the level and quality of care and comfort for the patient.

Providing the best possible care through the latest medical technology, minimizing the total time the patient resides in medical transit, and minimizing the number of times a patient is handled during this transit to an appropriate CONUS medical facility, are the primary objectives motivating research in the area of aeromedical evacuation. The problem of mass aeromedical evacuation of patients over long distances is unique to the military and has few parallels in the civilian sector. Military analysts have creatively attacked the problem using a variety of techniques. Most analytical work focuses on a particular segment of the system and seeks to determine the amount of resources required or an optimal allocation of a given set of resources in a defined scenario.

III. Computer Simulation Model

This chapter describes the methodology used to complete the first research objective presented in Chapter 1. This chapter includes two main sections. The first contains the AE scenario created by the sponsors AMC/XPY and medical planning experts in AMC/SG. The next section, model formulation, states the key assumptions and limitations made in constructing the simulation, describes each of the routines comprising the simulation code, and discusses the validation and verification techniques employed.

The objective of this research is to develop a flexible methodology that represents the key elements and policies affecting performance measures of strategic aeromedical evacuation and to apply appropriate statistical tools to better understand the relationship amongst these factors and policies.

To achieve this objective a modular approach was taken to develop the simulation code, with each module representing a particular process, or major element of strategic AE. In order to better respond to the natural "what-if" nature of a contingency planning environment, the model incorporates a data driven design. This allows the analyst to quickly examine an array of options under consideration by medical planners by means of editing the input data structure, not recoding the simulation. This modular, data driven philosophy guided the code development.

The desire to understand the general impact and interrelationships of the major strategic AE elements influenced the choice of statistical techniques to study the simulation output. A description of these techniques and the results they produced are found in Chapter 4. The purpose was not to perform a definitive analysis to determine a patient's mean time in system for a given scenario. Rather, the goal was, given a representative scenario, to take a macro approach to determine the major drivers affecting strategic AE and the fundamental relationship between these factors. This serves the analyst in validating the simulation, and it serves the medical contingency planning community by confirming or denying their intuition of the process, and providing the framework for better understanding of the possible tradeoffs amongst the key elements and policies for strategic AE.

Before setting the stage with a description of the scenario provided by the sponsors, it is important to understand the unique nomenclature that appears in this chapter. A characteristic strength of the SIMSCRIPT language is its "readability". This is primarily attributed to its capability to allow variable names to assume lengths greater than eight characters and its inherent English-like syntax structure. Therefore, to distinguish model variable names in the text, they will appear in a slightly different font type and may be separated by periods. For example, the variable that describes the mean time between batch arrivals of patients at a 3E medical facility is denoted mean.batch.interarrival.time in the section describing creation of patients. Also, module names appear in all capital letters to remind the reader of their relative level and function within the context.

Scenario

The methodology presented in this chapter is not scenario dependent. Rather, the methodology is designed to quickly accommodate and be used to evaluate many different scenarios. These scenarios may differ in the intensity of conflict, location and number of medical facilities, quantity of airlift and medical resources available, or AE strategy and policies employed.

It is beneficial, however, to use a representative scenario to exercise, evaluate, and to some extent validate the capabilities of the methodology. The following scenario, provided by the sponsors of this research, serves this purpose and also provides a baseline for analyzing the simulation output.

The scenario consists of a 180 day period of conflict fought in two separate theaters. The theaters, Southwest Asia (SWA) and the Far East, represent a situation which places a great demand on AE airlift operations since aircraft are flying in two separate directions from the CONUS with one of the destinations being approximately halfway around the world .

The SWA theater contains three APOEs that are each fed by two 3E facilities. The Far East theater has two APOEs that are also each fed by two 3E facilities (see Figure 3.1). This accounts for a total of five APOEs serviced by ten 3E medical facilities.

A total of 45 Boeing 767-200 series aircraft with a capacity of 102 patients are available for use. These aircraft are based on either the east or west coast of the United States. The aircraft fly routes that are permutations of the basing location, the en route refueling stop, the onload APOE, and the CONUS destination region. For this scenario, since each theater is basing aircraft in one location, flying through one en route refueling stop (Spain for the SWA theater and Alaska for the Far East theater), loading passengers at an APOE, and then returning to one of seven CONUS regions (as will be discussed later, one of these is a dummy region), this results in 35 different routes. An additional two routes are also used to allow for picking up casualties that have reached a time threshold at the APOE. Each of these latter routes services each of the CONUS regions. SWA has a total of 22 routes and Far East has a total of 15 routes. Since aircraft are all based in the CONUS (for ease of maintenance) every aircraft is able to fly every route.

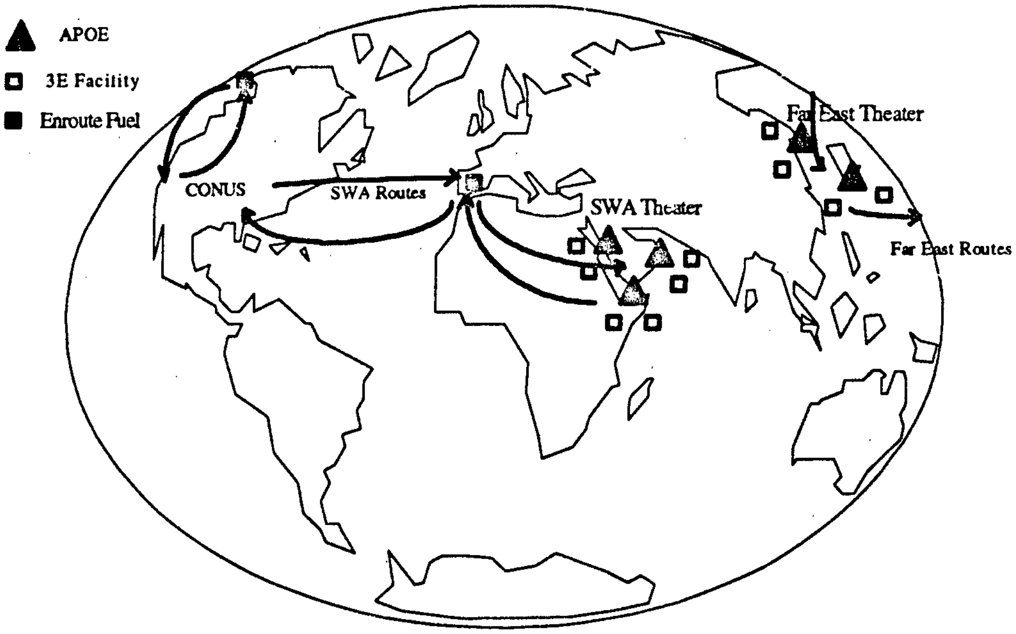

Figure 3.1. Two Theater Scenario

Casualties begin arriving on day one in the SWA theater and 40 days later they begin arriving in the Far East. Figure 3.2 shows how approximately 67,000 patients will arrive at the 3E facilities over the 180 day period. Figure 3.3 shows the breakdown of casualties and CONUS beds by type. Further casualty details are located in Appendix G.

It is interesting to note that one of the APOEs in the Far East theater will handle nearly 60% of the total casualties during the 180 day period, a disproportionate number.

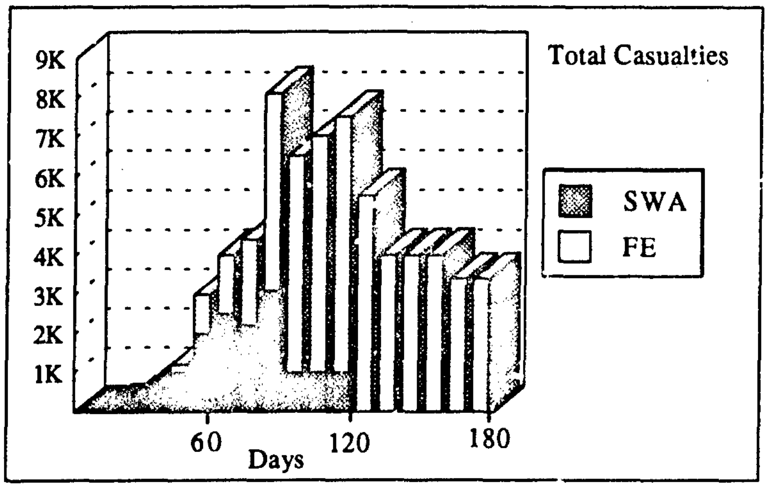

Figure 3.2. Two Theater Casualties

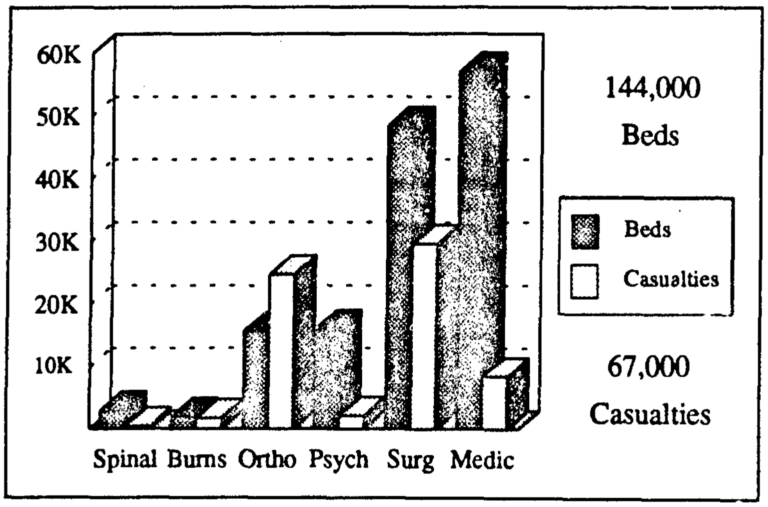

Figure 3.3. Two Theater Casualty Types

A total of 142,000 hospital beds are availabl^r in the six CONUS regions for patients. Figure 3.4 shows breakdown by organization.

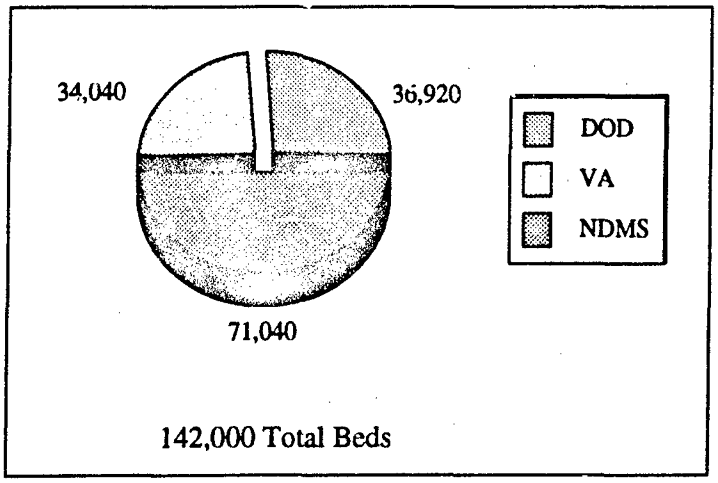

Figure 3.4. Two Theater Bed Availability

Each theater has a JMRO which communicates bed requirements to the ASMRO in the CONUS. For the baseline scenario, each JMRO will batch its bed requirements every eight hours. The ASMRO will regulate patients first to DOD beds, then to VA beds, and finally to NDMS beds. Each theater will fill each CONUS region using minimum flying distance as the priority. Cell fill policy is set to 90% and region fill policy is set to 80% for the scenario (5). A full explanation of these policies is found later in this chapter in the section describing event REGULATE.

Appendix C contains a detailed explanation of the aircraft, routes, locations, regulation policies, and bed availability. Also, the descriptions of each program module, found later in this chapter, further highlight the baseline scenario.

Model Formulation

One possible approach to introduce and understand the effects that random variables may have on the AE system is to construct a simulation model that mimics the currently planned strategic airlift plan. This is the most common technique used when faced with a complex problem in which it is not possible to use mathematical methods to obtain exact information on questions of interest (20:1). In fact, AMC/XPY, anticipating a simulation model might be the methodology, specifically requested the use of the SIMSCRIPT II.5 computer language. The organization currently has expertise and training in this language.

The intent of a computer simulation model is to mimic or imitate a real world process in order to more fully understand how it works and hopefully give decision makers the insight to make better decisions concerning its operation. While it is impossible to exactly represent any process, it is important to capture its major elements. This give and take between complexity and realism normally results in a set of assumptions that are made to simplify and therefore effectively use a simulation model.

Assumptions and Limitations. This research includes only the strategic operation of the Boeing 767 CRAF for medical evacuation. That is, the aircraft operations and patient movement from the designated aerial ports of embarkation in the theater of operations to the CONUS receiving hubs. It also assumes ample support personnel, flight crews, support equipment, etc. to sustain 767 operations and to handle casualties. The simulation controls the number of concurrent strategic flights to a particular third echelon facility by means of a resource called MOG, which is an acronym for maximum on ground. While the name implies ramp space allocated for parking aircraft, it can be used for the most limiting constraint at the 4E facility, which in fact may be the number of medical personnel available to on and offload an aircraft or the number of medical aircrews available to fly strategic missions. The analyst uses this variable as a throttle to control the scheduling of missions (while monitoring a variable which tracks the maximum and average number of aircraft on the ground at a 4E location at any given time).

The study will not consider the physical redistribution of patients in the CONUS once they have been delivered to a regional hub. However, it will track bed status by patient type for DOD, VA, and NDMS hospitals. No modeling of patient movement below the 3E level in the theater of operations is attempted. Therefore, movement of patients from the 3E facilities to a designated 4E facility is presumed to occur instantaneously. In other words, strategic missions are never delayed because of late arrivals from other areas within the theater of operations. The reason for this is to concentrate the study on the strategic element of the AE process, not its interaction with tactical theater airlift. These relationships and tradeoffs can be explored later if AMC/XPY expands the simulation to include CONUS redistribution and theater tactical airlift.

Structure. Fifteen different modules or routines, each performing one or more functions related to a major element of strategic AE or in support of model execution, and an input data file, make up the simulation model. Figure 3.5 provides an overview of how the fifteen modules are interrelated. The specific functions that each module accomplishes

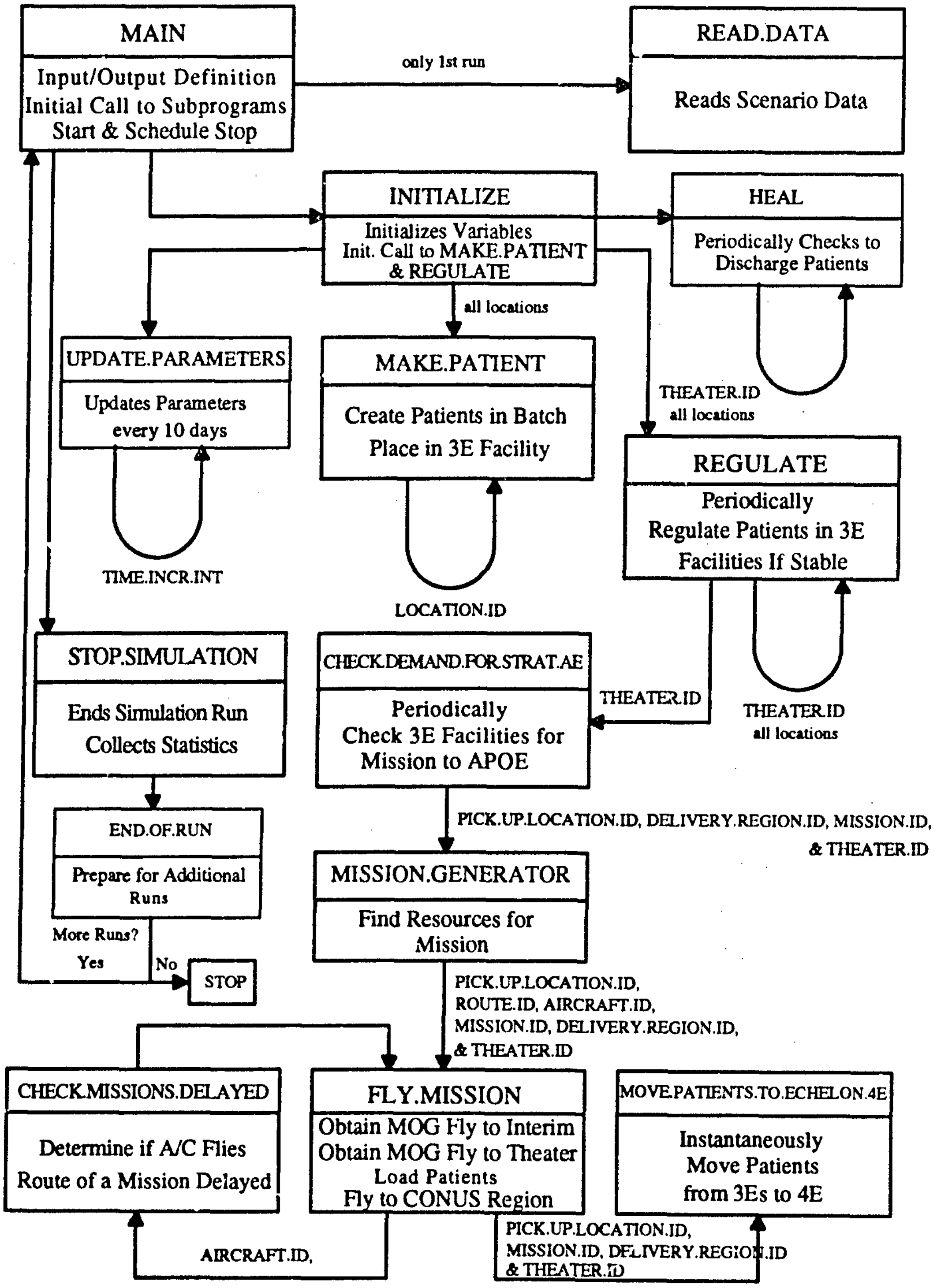

Figure 3.5. Master SIMSCRIPT Module Flow Diagram

are described in the following paragraphs. Appendix A contains the actual code for each of these modules. Appendix B contains the scenario or input information, which is read by the routine READ.DATA. Appendix C is an echo check of the data, written by READ.DATA. Appendix D contains the output from the baseline scenario experiment.

The simulation was written using the personal computer version of SIMSCRIPT II.5 computer language. The language is very portable and should require only slight input/output modifications to run in the Sun Micro environment at HQ AMC. To illustrate this, successful execution of this simulation on the Air Force Institute of Technology (AFIT) VAX mainframe computer required the removal of only four input/output statements found in the MAIN module and then subsequent recompilation.

PREAMBLE. The PREAMBLE module contains definitions for all the variables contained in the simulation, with the exception of strictly local variables, which are defined at the beginning of each module. Important features of the preamble include the definition of events, processes, temporary and permanent entities, sets (queues), integer, real and text variables as well as variables used for collecting statistics. Finally, the last operation within the PREAMBLE module sets the units of time to hours for this simulation.

MAIN. This module serves as the master control module for the simulation. Specifically, it defines the input and output files to use, makes initial calls to subprograms, starts the simulation, and schedules its stop time. In this case, MAIN first calls the routine which reads all scenario input data, READ.DATA, and then calls INITIALIZE to initially set the values of selected variables and arrays. The simulation uses the variable stop.time to schedule its termination. For the 180 day war scenario provided, this is 4320 simulated hours.

Routine to READ.DATA. This routine is divided into two main sections. The first part reads the scenario data from the file specified in MAIN. The second part delivers an echo print of all the variables that are read. This provides documentation of the parameters for each run, in addition to a check for possible input errors. There are five general areas available to print, each with a toggle variable found in the input data file (scenario.dat). Table 3.1 describes these options. Appendix C contains sample output.

Table 3.1. Echo Print Options

Topic	Description	Toggle Variable
Aircraft	Number, Capacity, Origination, Status (Idle or Busy)	aircraft.echo.on
Routes	Number, Name, Leg Information for Each Route (Leg Number, Origination, Destination, Flight Time, & Purpose), Aircraft/Route Assignments	route.echo.on
Locations	Number, Name, Mission Ramp Space for 4E Facilities, 3E Facilities Feeding 4E Patient Streams for 3E Facilities	location.echo.on
Regulation	Time to Begin Regulation, Regulation Frequency, Fill Policy for a Patient Type Cell Strategic CONUS Fill Policy	regulation.echo.on
Bed Status	Total Beds Available, Projected Occupied, and Occupied by Patient Type, CONUS Region, and Organizational Bed Type (i.e., DOD,VA, etc.)	bed.echo.on

The user must provide data for the aircraft fleet, flight routes, and locations. These may be output from an earlier deterministic technique used to size the problem. Estimates must also be made for the patient arrival rates and the distribution of patient types to each third echelon facility. This allows for the possibility of modeling casualties from different battle intensity levels. This is useful for representing separate campaigns that generate dissimilar types of casualties.

The simulation also identifies which aircraft can fly each particular route. This feature allows the analyst to examine the effects of different policy decisions regarding the AE concept of a single integrated manager. By assigning all aircraft resources to each route, the simulation represents central control over all strategic aircraft resources. The simulation can also represent decentralized control of aircraft to theater commanders by designating a portion of the total fleet to each set of routes within the jurisdiction of that commander. In this way planners can study the tradeoffs associated with dedicating a portion of the fleet to a particular route or making aircraft available across different routes.

Routine to INITIALIZE. This module performs two basic functions. First, it initiates event MAKE.PATIENT and event REGULATE. As its name implies, event MAKE.PATIENT generates casualty arrivals at each of the third echelon facilities. Periodically, as specified by the modeler, for a particular theater, event REGULATE finds a CONUS bed for every eligible patient in every third echelon facility. The first call to REGULATE occurs at the time contained in the variable begin.regulate.time and then periodically according to the variable regulate.frequency (both in hours). Subsequent calls to both of these events occur recursively. The second function of this module is to initialize variables and or arrays before execution of the simulation.

Event MAKE.PATIENT. The primary purpose of this module is to create the appropriate number and type of patients for the given scenario. Appendix F contains a flowchart showing how this module works. As previously noted, the module is first called by INITIALIZE and subsequently creates patients for each third echelon facility via a recursive call to the module. Each time, the event passes the location number of the 3E facility where the patients are created.

The time between arrivals for each batch of patients is presumed to have an exponential distribution. Each 3E facility has an attribute, mean.batch.interarrival.time, which contains the mean value for this distribution. This parameter may be changed periodically during the simulation by event UPDATE.PARAMETERS.

Each time the module is executed a batch size is determined by drawing .rom a uniform distribution and using the truncated or integer value as the number of patients arriving. Uniform distribution parameter values are also maintained as attributes of the 3E location. For this scenario, batch sizes at all 3E locations are assumed to be uniformly distributed between 5 and 25 with a mean value of 15 patients per batch. This spread represents the variability in the numbers of patients delivered to the 3E location from lower echelons. The range of possible batch sizes represents transportation ranging from ambulances to buses to C-130 aircraft.

This batch arrival scheme is described by Law and Kelton (20:409) and is known as a compound Poisson process. Explicit modeling of the tactical transportation of patients would provide better insight into the choice of values for the two distribution's parameters, and may suggest an alternative batch arrival scheme. Given, however, that the modeler wants to exercise a model that addresses only the strategic elements of AE, it is the responsibility of the analyst to properly batch patient arrivals in such way so as to achieve a specified expected number of casualties for the theater for a given period of time.

For example, suppose that during a ten day period, 2000 patients are expected to arrive at a given 3E facility in batches with a mean of 15 patients each. To determine the mean time between batch arrivals:

First convert the arrival rate into the appropriate units, e.g., hours,

$$\text{number of patients to arrive in 1 hour} = \frac{2000}{10}\frac{\textit{patients}}{\textit{days}} \times \frac{10}{240}\frac{\textit{days}}{\textit{hours}} = 8.3333\ \frac{\textit{patients}}{\textit{hour}}.$$

Second, if patients were to arrive individually (in batches of size one) this would correspond to a mean time between arrivals of

$$1 / 8.3333 = .1200 \text{ hours.}$$

Third, since patients arrive in a mean batch size of 15, the mean time between batch arrivals is thus

$$.1200 \times 15 = 1.800 \text{ hours.}$$

The two-theater scenario calls for this number of casualties across six APOEs in the SWA theater between days 50 and 60 of the war (see Appendix E). Note the above value. 1.80, is multiplied by the number of APOEs in the theater, 6, to obtain the value, 10.8000 to place into the variable mean.batch.interarrival.time for the 4E locations.

Finally, this module then assigns values to the attributes for each patient for use later in the simulation. These attributes include the time the patient arrived at the 3E facility, the type of patient (determined by the random step variable patient.type.mix for each location), the time the patient is stabilized (since a patient must be stabilized before he or she may be regulated to CONUS hospital), the patient's regulation status, and the patient's heal time (which will eventually result in the patient's removal from the CONUS hospital). Distribution of patient type, and their associated stabilization and heal times are found in Table 3.2. These estimates were provided by AMC/SG. For the provided scenario, all 3E locations generate the same distribution of patient types. Medical planners use the medical

Table 3.2. Patient Type Parameters (25, 6:7)

Patient Type	Code	Probability this type	Mean Time to Stabilize (hrs)	Mean Heal Time(days)
Medicine	1	.126	6.0	16
Surgery	2	.441	6.0	29
Psychiatric	3	.032	6.0	24
Orthopedic	4	.368	12.0	50
Burns	5	.026	12.0	33
Spinal	6	.007	24.0	38
OB/GYN	7	.000	-	-
Pediatrics	8	.000	-	-

planning module (MPM) to project the number of casualties expected given the scope of anticipated combat operations. History has shown that approximately 40 per 1000 combatants will require hospitalization per day (32:7).

Event REGULATE. This event performs the medical regulation function for each theater of operations. Appendix F contains a flowchart of this module. For the specified theater, this module regulates every eligible (stabilized) patient in every 3E location each time it is called. The first calls to this event occurs from MAIN at a time specified by the values found in the array variable begin.theater.regulate. The event then calls itself periodically (every theater.regulate.frequency hours).

This event offers the modeler two very different ways to assign patients to medical facilities within the CONUS. This option, specified by the variable strategic.fill.policy, is either set to "region.then.organization" or "organization.then.region". If the latter is chosen, the program will first attempt to fill all beds within a given organization type (e.g., DOD, VA, or NDMS) for a particular patient type across all regions. Once the organization type is filled, the routine searches the next organization across every region, and so on. If the variable is set to "region.then.organization", the search for a bed for a given patient type occurs first within a region across all organization types. Once a region is full, the search continues in the next region. Current policy is to fill within the organization type first, or "organization.then.region" as annotated in the model. Of obvious interest is the difference this policy makes for the time in system for patients, since it could result in strategic aircraft overflying VA & NDMS beds in a region relatively close to the theater in order to first fill DOD beds in other regions.

This module uses three separate 3-dimensional arrays to track the available, projected occupied, and occupied beds for each type of patient in each type of organizational facility, in each region in the CONUS. Since there are eight types of patients, four organizational types, and seven CONUS regions, a patient will be assigned or regulated to one of 224 individual cells (see Figure 3.6). The analysts may designate a maximum level to fill each of these cells. The variable cell.fill.policy specifies this value and is presumed to apply to all 224 cells. This controls the workload across available CONUS facilities and prevents the regulator from bringing medical capabilities at some facilities to maximum capacity while those at other facilities are idle.

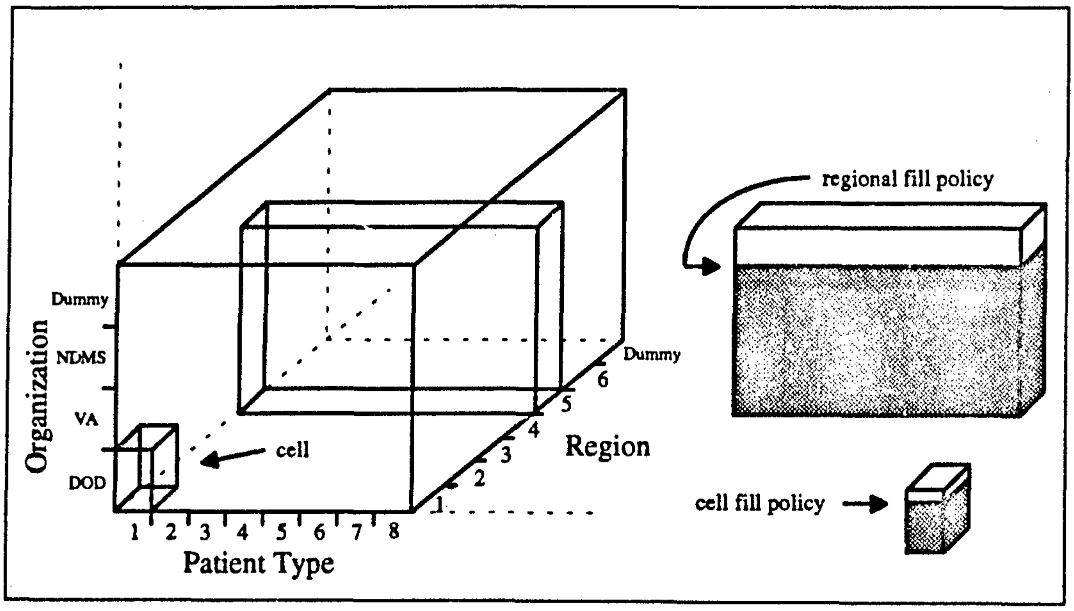

Figure 3.6. Three Dimensional Representation of Bed Availability/Occupancy

Each CONUS region also maintains a fill status attribute. This allows the regulating process to skip a region entirely or cease regulating to a region once it reaches a specified fill level. This is needed to control the amount of demand placed on medical resources for a given region.

Note that the fourth organization type and the seventh region are dummy parameters. If patients are regulated to cells containing these indices, the modeler should increase the fill policies. If after maximizing these policies, these cells continue to collect patients, a bed shortage has been identified.

Every patient regulated during this event is assigned a regional destination and his regulation status is updated to "regulated". Each time this theater regulation takes place, a call is made to CHECK.DEMAND.FOR.STRAT.AE to determine if there are sufficient numbers of patients regulated to each region to warrant scheduling strategic AE missions.

Event CHECK.DEMAND.FOR.STRAT.AE. This module, as its name implies, checks the demand for strategic AE for every 4E facility within a specified theater (see Appendix F for a flowchart). For each 4E facility, this module queries every 3E facility which may feed it patients, and performs a cumulative count of the number of patients that have been regulated to each CONUS region. If enough patients have been regulated to a particular region to fill a strategic aircraft, the event calls event MISSION.GENERATOR passing the identification of the 4E facility which desires the mission, the destination (CONUS region) of the mission, and a unique mission number.

When the event locates enough patients to fill an aircraft, these patients' attributes are updated for later use in the simulation. The patient's regulation status is changed from "regulated" to "regulated.and.mission" signifying the patient has been assigned against a specific mission number. This number is stored in the patient's attribute sae.mission.

This module implements a key assumption of this simulation, that is, strategic AE missions will be demand driven, not flown according to a routine schedule. If however, the modeler wanted to incorporate a routine schedule, this could easily be accommodated by an initial call to MISSION.GENERATOR with periodic (according to the schedule) recursive calls. The modeler would also need to incorporate a strategy for patient selection and mission sequencing.

This simulation uses this demand scheme based upon the experienced recommendation of the AMC Analysis Group. This assumption recognizes that AE, just as any type of airlift, serves the commanders in the field and therefore must respond to their needs.

Event MISSION.GENERATOR. Once there is sufficient demand to warrant a strategic AE mission, CHECK.DEMAND.FOR.STRAT.AE calls this module, passing several input parameters. Specifically, the parameters are the 4E location where the aircraft must pick-up patients, the region in the CONUS to deliver these patients, a unique number to identify and track the mission, and the theater of operations. The purpose of event MISSION.GENERATOR is to use these inputs to find a specific aircraft and route for the mission.

The event will always find a route for the mission, however, sometimes a plane will not be available. Based on this, there are two courses of action the event takes. If an aircraft is found, this information, along with the pick-up location, route number, and mission number, are passed to process FLY.MISSION. Once an aircraft is found its status is changed to "busy" and it is committed to fulfilling that specific mission. If an aircraft is not found, then this is noted by creating a mission delayed (temporary entity) and storing the pick-up location, route number, mission number, and theater (as attributes of this mission delayed) for future reference (the temporary entity is filed in the set mission.delayed.pool).

Process FLY.MISSION. Once a pick-up location, route, aircraft, mission, and delivery region are pinpointed, this information along with the theater identification is passed to this module. Process FLY.MISSION performs several functions. First, the process waits, representing an administrative preflight processing time. Then it requests a unit of maximum on ground (MOG) resource at the location of the 4E facility where it will pick up its patients. After it receives this resource, it immediately calls MOVE.PATIENTS.TO.4E, which will identify the patients (using patient attribute sae.mission) in every 3E facility with patients manifest for this mission and instantly place them in the 4E facility for pick-up. Then the aircraft travels the remaining legs of the route designated for this mission. Travel leg attributes such as flight time and reason for stop are all read in READ.DATA. Travel legs, denoted as travel.leg in the code, are temporary entities stored in a set called route.leg.sequence. Each route has (owns) an associated route.leg.sequence.

At the end of each leg, one of four options is exercised based on the reason for stop. If the purpose of the stop is to load patients, then the patients marked with this mission, are removed from the location's patient list (set) and then loaded onto the aircraft (placed in the aircraft's manifest list, also a set that every aircraft owns). If the purpose is to unload patients, the patients are removed from the aircraft's manifest list and placed in the CONUS region's patient list. If the purpose is to end the mission, it removes any patients which may be left on the aircraft and places them in the current location. The last option is for the aircraft to stop for refueling.

When the mission is complete, FLY.MISSION waits a period of time to reconstitute the aircraft for another mission. After this delay several statistics are updated and a call is made to event CHECK.MISSIONS.DELAYED passing the aircraft's identification. If the aircraft can fly any mission which has been delayed, an immediate call back to FLY.MISSION is made, passing the attributes stored in the mission.delayed entity along with the aircraft identification. In the case where the aircraft is not needed to fly a delayed mission, the aircraft's status is changed to "idle" making it eligible to fly future missions.

Process MOVE.PATIENTS.TO.ECHELON.4E. This module is a very simple routine called by FLY.MISSION. Its purpose is to search every 3E facility which needs to transport patients to the 4E facility identified as the pick-up location for a specified strategic mission. The module identifies the patients using the patient attribute sae.mission, removes the patient from the 3E locations' patient.list (set), and files the patients in the 4E location patient.list. (Remember that the simulation does not model tactical AE explicitly, therefore this travel time is modeled as instantaneous.)

Event CHECK.MISSIONS.DELAYED. This event attempts to match an available aircraft resource against a pool of delayed missions. These missions have been previously identified by MISSION.GENERATOR as not having an available aircraft to perform the mission. Each time an aircraft completes a mission this event is called by FLY.MISSION, passing the aircraft identification. If the aircraft can fly a delayed mission, the mission is started by passing the needed attributes back to FLY.MISSION. The temporary entity, mission.delayed, is then destroyed. If there is no match, the aircraft's status is updated to "idle" and it becomes eligible for future missions.

Event HEAL. This event performs a very simple but important function. Periodically, daily in the provided scenario, it checks within each CONUS region and determines which patients are ready for discharge thus freeing the bed for regulation. It does this by comparing the patient's heal.time (assigned according to patient category) to the current time. The analyst specifies the time to begin this event, begin.heal.time, and the frequency to call the event, heal.time.frequency, in the input data file. Patients that are regulated and delivered to the dummy region are never healed since patients remaining in this region at the end of the simulation define the total bed shortage over the entire simulated period. To obtain a breakdown of when these shortages occur the analyst should add a print of the desired information to the UPDATE.PARAMETERS module.

Process STOP.SIMULATION. This process signals the end of one replication of the simulation. It is scheduled by MAIN. The purpose of the module is two-fold. First it stops the simulation after a specified time and second it collects numerous statistics of interest. This process ends by making a call to the routine END.OF.RUN where the simulation resets all necessary parameters to make another replication. If all replications have been accomplished it prints the grand statistics for all runs.

Routine to END.OF.RUN. This routine, called by STOP.SIMULATION, prepares the simulation for another replication. The SIMSCRIPT language requires the programmer to reset all variable, array, and counter values, destroy all entities and resources, and remove all scheduled events and processes from the simulation calendar. The analyst should note this requirement when making future modifications to the code. The user should also take care not to destroy entities and value settings which do not change over replications. For example, the analyst does not want to destroy the temporary entities travel.leg and aircraft.servicing.a.route since they contain important route and aircraft information that is read in only once at the beginning of the simulation and does not change with each replication. On the contrary, if the analyst failed to destroy the temporary entity mission.delayed, this would carry over to the next replication and cause an additional mission to be flown. The point to remember is while this overhead is relatively easy to accomplish, it can introduce subtle errors if not given close attention.

Event UPDATE.PARAMETERS. This event provides a way to update any parameters that may change over time during the simulation. For example, for the scenario provided, casualty arrival rates change every ten days. This module is called after ten days of simulated time, updates the mean.batch.arrival.time for each 3E location, and the schedules itself to occur again in ten days. This module can also be used to print intermediate results. Again, for the given scenario, every ten days this routine prints the average utilization rate for the strategic aircraft during the past ten days only.

Verification and Validation. The process of verification and validation is paramount to a model's eventual acceptance and use. This process is continuous and remains fundamental to a model's influence and utility over its entire lifetime. This section describes what steps have been taken thus far and what should be emphasized in the future regarding verification and validation.

First, Law and Kelton define verification as "determining that a simulation computer program performs as intended, i.e., debugging the computer program" (20:299). Validation is defined as "determining whether the conceptual simulation model (as opposed to the computer model) is an accurate representation of the system under study" (20:299). They also offer several techniques to guide both verification and validation. For verification, some of these techniques include modular program development, "structured walk-throughs", sensitivity checks of the output, trace of variable values, and the use of established simulation packages (20:302-306). For validation, Law and Kelton describe a three step approach for model validation that includes developing the model with "high face validity", testing the assumptions of the model empirically, and finally determining how representative the output from the simulation is compared to the real system

(20:308-314). These techniques for verification and the three step approach to validation provide the framework for the remainder of this discussion.

The first and most important step for any analyst is to develop a comprehensive understanding of the problem. This was best accomplished by face-to-face meetings with the eventual exerciser of the simulation, HQ AMC/XPY, and the eventual user of its results, the AMC Surgeon's staff along with several other AE system related experts. The methodology developed so far has relied on an iterative process of coding and review that involves the medical staff, the analysis group, and the analyst. This ensured the simulation contained the essential parameters which may bear on decisions later.

Since the simulation code has just been developed it follows that most of the effort so far has been focused in the area of verification. Several measures have been taken to ensure the simulation code works as desired. The simulation language, SIMSCRIPT, cannot be considered "user-friendly" in helping the analyst verify the code. Complexity and overhead are by-products of the language's power and readability. In short, SIMSCRIPT requires the programmer to create code to obtain most parameter information and checks during debugging.

However, the first verification technique of programming in modules was well supported by the language. The code was primarily developed using the personal computer version of SIMSCRIPT. This version directly supports development by module to include separate compilation of modules. The author wrote each module sequentially and exercised it using a scaled down version of the scenario. Checking to make sure that each module performed its function and passed the correct values of parameters to other modules helped to verify that the entire code was performing its purpose. The building of code in modules that represented actual AE processes helped to quickly identify and correct logic problems. In addition, this structure lays the foundation for ease of maintenance and updates to the code in the future. It also establishes logical points of connection if the code is ever expanded to include CONUS redistribution of patients or tactical AE below the 3E level in the theater.

The next technique, the structured walk-through, is simply having a group of peers review the code for correctness and efficiency of approach. This avoids single-mindedness and inefficiencies in the structure of the code. While more than one person has reviewed a majority of the code, it was written in an academic environment which naturally precluded a thorough peer scrub. One of the first tasks for the recipient of this model should be a rigorous line-by-line review of the code.

The third verification technique is to continually check the output of the model for reasonableness. This was done by liberally imbedding print statements throughout the code to check parameter values and counts for specific intermediate and summary time periods. (These print statement do not appear in the code provided in Appendix A for readability, however, an unsanitized version will be provided to AMC/XPY). Initially, stochastic representations of events were made deterministic to ensure logic was correct. Before stochastic representations were implemented a reasonableness check was made to make sure the stochastic process was represented correctly.

For example, for the creation of patients (described earlier in this chapter), the analyst first used a deterministic scheme to create a known number of patients. This made it easier to work through the logic of the routine. After the logic was established the analyst incorporated the stochastic arrival of batches of patients. Given the parameters of the distributions, the exponential for the time between arrivals of batches of patients, and the uniform for batch sizes, an estimate of the expected value of the total number of patients was made for the simulation run length. Comparison of the theoretically expected number of patients given the distribution parameter values versus the actual number the simulation created showed a difference of about 2 percent. Considering the large variance associated with the exponential distribution the results were deemed acceptable. The analyst practiced this type of approach within each of the modules.

Another useful technique that unfortunately was often used during the code's development was tracking specified variable values over time to ensure the correct information was passed between modules. Law and Kelton refer to this technique as a "trace" (20:303). This approach proved invaluable in debugging this particular code because of the importance the logic places on maintaining and transferring parameter values between modules. Refer again to Figure 3.5, Master SIMSCRIPT Module Flow Diagram, and note the number and type of information passed between modules.

Finally, the simulation was written in a well established simulation language. This precluded the requirement to write and vigorously verify code for such items as probability distributions, statistical collection and random number generation. Still, this simulation language was used with a watchful eye. For example, even though the code uses statistical features inherent in the language to collect and print a grand mean and standard deviation over several replications, it became apparent that the standard deviation provided came from a biased (low) estimate of the variance. This value was not used in the univariate analysis described later in the next chapter. Rather, an unbiased estimate of the sample variance was computed and used. Mendenhall, Wackerly, and Scheaffer provide a complete a explanation of the difference (26:304-315).

Other verification techniques such as animation are gaining in popularity but the structure of the approach (data driven) and the time required did not allow pursuit of this approach. Overall, the code is more verified than validated, but verification still warrants attention in the future.

Validation of a computer simulation model that attempts to represent a process that has never occurred (as it is currently foreseen and planned for) provides a formidable task. Since this model is emerging from infancy and will continue to mature it would be foolhardy to proclaim the model "validated". However, the analyst has aggressively pursued the three step approach for model validation described by Law and Kelton, which they adapted from Naylor and Finger (20:307).

The first validation step, referred to as gaining "high face validity" (20:308), describes how this research began. There have been two face-to-face meetings with both the end user, the AMC medical planning staff, and AMC/XPY, the organization who will inherit and exercise the tool. These meetings with the "system experts" (20:308) produced the framework and assumptions for the simulation model. In addition, dozens of other telephone conversations with these and other experts in closely related fields, such as the staff at the ASMRO, has helped to define reasonable assumptions about model fidelity, values for input data, and model logic. The experience and intuition of these experts, key factors mentioned by Law and Kelton (20:309), have played heavily in defining important assumptions for the model's structure. As mentioned earlier in this thesis, the key insight for modeling the AE process as "demand driven" was suggested by Litko, head of the AMC Analysis Group, based on his experience. This ingredient of validation can only improve when ownership of the codes transfers to AMC/XPY, who are collocated with and work closely with the medical planners on a variety of issues.

The second step toward validating a computer model is to test its assumptions empirically. The primary tool used to accomplish this was a preliminary sensitivity analysis. This provided a quantitative way to test whether or not the simulation responded in the way expected when a single factor or policy was changed. The principal response observed was time in system for a patient. The results of this sensitivity analysis also helped the analyst select the factors (and their magnitude) to use in the designed experiment discussed in the next chapter.

Law and Kelton stress that when conducting a sensitivity analysis it is essential to use the method of common random numbers, a variance reduction technique, to control the amount of randomness in the simulation (20:311). The end objective of common random numbers is to allow comparison of different alternatives or policies "under similar experimental conditions" in order to gain confidence that differences in performance (patient time in system for this case) are due to changes in the policy and not due to random fluctuations of the experimental conditions (20:613-614). The aim is to reduce the amount of variance associated with an output variable "without disturbing its expected value" (20:612). The SIMSCRIPT language makes implementation of the method of common random numbers quite simple, since it provides ten separate random number streams for use. Each separate random variate in the simulation was assigned a separate stream number. If several random variates occurred within a single process (such as FLY.MISSION), they were all assigned the same stream number. Measuring the effect of common random numbers is difficult (20:615), but as evidenced in the measures of the variance for time in system during sensitivity runs, it seems to give the desired effect.

The last validation step is to examine whether or not the simulation output is representative of the real world (20:311). Unfortunately, there is no real world process to measure in this case, so one must rely on the intuition of what experts in the field think are representative. This step of the validation process is best addressed through the factor analysis described later in the next chapter. The factor analysis is well suited for this task because it is a data reduction technique that seeks to provide insight to the underlying process expressed through the chosen vector of simulation output. By performing a factor analysis on simulation data generated from a designed experiment, the analysts can identify key factors and their relationships. These insights can then be compared against the insights and intuition of the system experts.

Verification and validation of computer simulation models is an ongoing effort. In many cases the validation effort involves more art than science, as is evident in interpreting the factor score plots found in Chapter 4, and then attempting to assign meaning to them. Nevertheless, an effort has been made to exercise the techniques advocated by Law and Kelton and widely accepted by many using simulation to assist the decision process.

IV. Analyses

This chapter presents the analyses and findings of this research. Recall that this research had two primary objectives. The first was to construct and document a computer simulation model that addressed the major elements of strategic AE. The second objective was to use the model to investigate a representative scenario provided by the user. Both these objectives have been achieved.

This chapter primarily describes two broad approaches used to examine the simulation output. The first, labeled univariate analysis, seeks to determine the effect different policy choices or resource constraints have on average patient time in system. The second, multivariate analysis, examines the multiple output variables searching for underlying factors and their interrelationships. This type of analysis serves to validate the methodology and unveil system insights and possible tradeoffs decision makers should know exist.

One must apply caution not to draw specific conclusions about AE operations based strictly on the results of these two analyses, remembering they are based on a single, two-theater scenario, where one of the APOEs receives a disproportionate number of the total casualties. However, one can certainly reach some broad conclusions about how AE polices and resources are interrelated.

Before any analysis can take place however, there must exist data. A brief description follows on what output measures were initially thought important to measure, and the sensitivity analysis and resulting designed experiment used to obtain this data.

Measures of Effectiveness

The model captures several important measures of effectiveness (MOEs) including,

1) Average time a patient is in the system (time in system measured from the time a patient is stabilized and is eligible for strategic AE to the time the patient arrives at the CONUS region).

2) Average time in system for each of the two theaters, Far East and Southwest Asia.

4) Average utilization rates for each type aircraft.

5) Maximum utilization rates for each type aircraft over the length of the conflict (measured every ten days).

6) Average number of patients in all 3E facilities over the length of the simulation.

7) Average number of aircraft parked at 4E facilities during the simulation.

8) Total percentage of patients transferred to the CONUS during the simulation.

9) Percentage of total missions that were delayed because there was not an aircraft available to fly the mission.

These measures of effectiveness were the primary output values recorded during the sensitivity and designed experiment runs (see Table 4.1 and Table 4.3). There are many other values that are captured by the different print echos which are not listed above. Refer to Appendix D for examples. Of course, just about any parameter of interest can be recorded by the simulation with further modification of the code.

Sensitivity Analysis and Designed Experiment

As mentioned in Chapter 3, performing a sensitivity analysis serves as part of the validation process. Specifically, the analysis provides a way to test the model's assumptions empirically. In this way, the analyst was able to quantitatively check the effects of changing major policies or resources. Through interacting with the user and experiencing the process of structuring the simulation code to represent aeromedical concepts, the analyst began to acquire an intuition for what major input factors were important. The following are the major inputs the analyst was interested in experimenting with after the model was constructed:

- Frequency of the regulation process for each theater
- Strategic regulation policy (whether patients were regulated to organizations first or to CONUS regions first)
- Number of Boeing 767 aircraft available
- Command and control structure of this fleet (centralized or decentralized)
- Resources available at the APOE to service both patients and AE operations (referred to as MOG, maximum on ground in the code).

In order to compare alternative policies, the analyst formed a baseline according to known policies and resources as well as recommendations from system experts. The following baseline (which is run 2 in Table 4.1) was established: a regulation frequency of 8 hours for each theater, a strategic regulating policy of filling first by organization and then by region (all DOD beds filled first across all regions, then VA beds, etc.), a total of 45 available aircraft which could be shared across the two theaters (that is, centralized command and control). Finally, the MOG resource was set at 3 for each APOE. This meant each APOE could service a maximum of 3 aircraft simultaneously. An aircraft attempted to seize this aggregated resource after refueling at the interim enroute location and relinquished it after loading its patients at the APOE.

With this structure in place all that was left to do before making sensitivity runs was determine the number of replications to perform for each run. With two goals in mind, first, obtaining enough precision in the measurement of ıverage time in system for a patient to determine if there was a significant difference among policies and second, keeping the amount of central processing unit (CPU) time at a reasonable level, an estimate was made for the number of replications needed.

The baseline case was run for 25 replications to obtain a grand mean and variance for each output measure of interest. (As a note of interest, the simulation took approximately one minute to compile and approximately four minutes per replication to execute on the VAX mainframe computer, or a little more than one and half hours for the baseline case.) For the 25 observations, a mean of 73.8 and a sample variance of 2.39 resulted for a patient's average time in system. Thus, it took about three days on average to transport a patient from the theater to the CONUS region. It then seemed reasonable to establish the number of replications as that number which would result in a high confidence (99 percent) that our estimate of the expected time in system would have an absolute error of estimation of less than three hours. From statistics, it is known that approximately 99 percent of the sample means will lie within three standard deviations of the population mean in repeated sampling. Thus, to obtain the number of samples required one need simply to find n such that,

$$\frac{3\sigma}{\sqrt{n}} = 3$$

It follows that,

$$n = \sigma^2 \ .$$

Since the (sample) variance for the 25 baseline replications was 2.39, this suggests that at least three replications are required. However, not knowing how the variance might change as the policies and resources are changed, a decision was made to use 5 replications (which implies a reasonable average of 20 CPU minutes per run). Five replications turned out to be the highest number used even though the variance for runs with a MOG value of 2 produced sample variances around 10-11 which would suggest the need for approximately fifteen replications. This did not affect conclusions drawn about differences in policy however since the high variances, due to the lack of MOG resource, resulted in significantly higher times in system.

With the previously described five major inputs and the principal output measure in mind, a sensitivity analysis was performed to determine the effect on time in system by varying each of the input variables across a range of values. For the most part, each sensitivity run varied in only one input parameter. However, sometimes a second factor would also be changed. Table 4.1 contains a complete listing of the input settings and output generated.

Table 4.1. Sensitivity Runs (5 replications at each run)

	Input Parameters					Output								
Run	Regulation Frequency (hrs)	Regulation Policy	# of Planes	Cmd & Control	Max on Ground	Time in Sys	TIS SWA	TIS FE	Avg Ute hrs/day	Max Ute hrs/day	Avg # in all 3Es	Avg A/C at 4E	% to CONUS	% Msns Delayed
1	4	org.then.reg	45	central	3	68.1	95.6	60.6	2.6	5.1	77	0.141	98.4	0.0
2	8	org.then.reg	45	central	3	73.1	104.2	64.5	2.5	5.0	86	0.126	98.2	0.0
3	12	org.then.reg	45	central	3	79.8	125.8	67.1	2.4	4.9	98	0.121	98.1	0.0
4	16	org.then.reg	45	central	3	83.5	132.7	69.9	2.4	4.9	104	0.119	98.2	0.0
5	24	org.then.reg	45	central	3	87.4	139.9	73.1	2.4	4.7	121	0.116	97.8	0.0
6	36	org.then.reg	45	central	3	92.9	131.6	82.4	2.4	4.7	139	0.114	97.1	0.0
7	48	org.then.reg	45	central	3	99.8	135.7	90.0	2.4	4.7	153	0.114	97.1	0.0
8	8	org.then.reg	20	central	3	73.7	105.1	65.0	5.6	11.1	87	0.126	98.1	0.0
9	8	reg.then.org	20	central	3	56.6	73.9	51.8	5.3	10.7	64	0.125	98.6	0.0
10	8	reg.then.org	45	central	3	56.2	73.1	51.5	2.3	4.7	63	0.125	98.6	0.0
11	8	org.then.reg	45	decentral	3	74.6	109.4	64.9	2.5	4.9	89	0.126	98.2	0.0
12	8	reg.then.org	45	decentral	3	57.0	73.6	52.4	2.3	4.7	64	0.126	98.6	0.0
13	8	org.then.reg	20	decentral	3	75.0	108.4	65.8	5.6	10.7	90	0.126	98.3	12.8
14	8	org.then.reg	15	central	3	75.0	106.4	66.3	7.6	15.2	89	0.126	98.3	16.7
15	8	org.then.reg	20	central	3	73.7	105.1	65.0	5.6	11.1	87	0.126	98.1	0.0
16	8	org.then.reg	25	central	3	73.7	105.7	64.8	4.5	8.9	87	0.126	98.6	0.0
17	8	org.then.reg	30	central	3	73.1	104.2	64.5	3.7	7.5	86	0.126	98.2	0.0
18	8	org.then.reg	35	central	3	73.1	104.2	64.5	3.2	6.4	86	0.126	98.2	0.0
19	8	org.then.reg	40	central	3	73.1	104.2	64.5	2.8	5.6	86	0.126	98.2	0.0
20	8	org.then.reg	15	decentral	3	116.8	110.0	118.7	7.4	13.2	156	0.126	98.2	54.8
21	8	org.then.reg	45	central	2	108.4	109.7	107.9	2.5	4.4	141	0.165	98.2	0.3
22	8	org.then.reg	45	central	4	73.8	108.9	64.1	2.5	5.0	88	0.125	98.2	0.0
23	8	org.then.reg	45	central	5	73.4	106.4	64.2	2.5	5.0	87	0.125	98.3	0.0
24	8	org.then.reg	45	central	6	73.4	106.7	64.1	2.5	5.1	87	0.125	98.2	0.0

One of the purposes of the sensitivity analysis was to obtain a relative feel for the effect of factors and how their values affect time in system. Factors that were significant, (hopefully all of them since the effort had been made to model them), would be used in a designed experiment.

The purpose of the designed experiment was to determine how changes in one or more of the major factors affect the vector of output measures. For the univariate analysis, time in system was the measure of interest. For the multivariate analysis, all the output measures were initially considered. Table 4.3 shows the structure and results of a full 2^5 factorial design (all main factors were shown significant). Each design point, or simulation run, consisted of 5 replications. Selection of the factor levels was based on a combination of the results of the sensitivity analysis and real-world constraints or recommendations from system experts. Table 4.2 shows the high and low values selected.

Table 4.2. Factor Level Settings

Factor	High	Low
Regulation Frequency	24 hrs	8 hrs
Regulation Policy	Organization first	Region first
Number of Aircraft	45	15
Command & Control	Decentral	Central
Max on Ground (MOG)	4	2

Table 4.3. Designed Experiment
(5 replications at each run)

	Input Parameters					Output								
Run	Regulation Frequency (hrs)	Regulation Policy	# of Planes	Cmd & Control	Max on Ground	Time in Sys	TIS SWA	TIS FE	Avg Ute hrs/day	Max Ute hrs/day	Avg # in all 3Es	Avg A/C at 4E	% to CONUS	% Msns Delayed
1	8	org.then.reg	15	central	2	116.8	119.7	115.4	7.6	13.4	154	0.145	98.1	54.2
2	8	org.then.reg	15	central	4	74.6	107.4	65.5	7.5	15.4	89	0.125	98.3	14.4
3	24	org.then.reg	15	central	2	121.1	147.4	113.7	7.3	13.5	174	0.121	97.6	48.8
4	24	org.then.reg	15	central	4	86.8	140.4	72.1	7.3	14.6	120	0.115	97.7	11.8
5	8	reg.then.org	15	central	2	93.7	82.6	96.8	7.2	13.4	122	0.146	98.5	51.5
6	8	reg.then.org	15	central	4	57.2	74.5	52.4	7.1	14.5	65	0.125	98.6	10.1
7	24	reg.then.org	15	central	2	98.8	95.1	99.7	6.9	12.9	133	0.121	98.4	45.4
8	24	reg.then.org	15	central	4	66.0	88.4	59.7	6.9	14.2	81	0.115	98.4	8.8
9	8	org.then.reg	45	central	2	108.4	109.7	107.9	2.5	4.4	141	0.165	98.2	0.3
10	8	org.then.reg	45	central	4	73.8	108.9	64.1	2.5	5.0	88	0.125	98.2	0.0
11	24	org.then.reg	45	central	2	15.3	138.5	108.9	2.4	4.3	165	0.128	97.6	0.0
12	24	org.then.reg	45	central	4	86.0	138.1	71.7	2.4	4.8	120	0.115	97.6	0.0
13	8	reg.then.org	45	central	2	85.6	72.6	89.2	2.3	4.4	109	0.161	98.5	0.0
14	8	reg.then.org	45	central	4	56.1	73.2	51.4	2.3	4.7	64	0.125	98.5	0.0
15	24	reg.then.org	45	central	2	93.9	89.0	95.2	2.3	4.2	124	0.128	98.4	0.0
16	24	reg.then.org	45	central	4	65.6	89.1	59.1	2.3	4.5	79	0.115	98.6	0.0
17	8	org.then.reg	15	decentral	2	155.4	109.1	168.4	7.4	12.9	217	0.130	97.7	65.6
18	8	org.then.reg	15	decentral	4	112.5	111.2	112.8	7.4	13.4	149	0.125	98.2	54.1
19	24	org.then.reg	15	decentral	2	152.2	140.0	164.4	7.2	12.7	233	0.116	97.4	62.9
20	24	org.then.reg	15	decentral	4	122.1	140.3	117.1	7.2	12.9	175	0.115	97.8	50.6
21	8	reg.then.org	15	decentral	2	108.8	75.7	117.9	7.0	12.9	147	0.130	98.5	57.2
22	8	reg.then.org	15	decentral	4	75.1	75.6	74.9	7.0	12.9	94	0.125	98.5	43.7
23	24	reg.then.org	15	decentral	2	113.3	90.7	119.5	6.8	12.4	156	0.116	98.4	54.4
24	24	reg.then.org	15	decentral	4	82.8	90.7	80.6	6.8	12.5	108	0.115	98.4	40.6
25	8	org.then.reg	45	decentral	2	107.3	109.1	106.8	2.5	4.5	139	0.164	98.2	16.1
26	8	org.then.reg	45	decentral	4	73.8	108.9	64.1	2.5	5.0	88	0.125	98.2	0.0
27	24	org.then.reg	45	decentral	2	114.2	138.9	107.3	2.4	4.4	160	0.127	97.6	12.8
28	24	org.then.reg	45	decentral	4	86.0	138.1	71.7	2.4	4.8	120	0.115	97.6	0.0
29	8	reg.then.org	45	decentral	2	86.9	73.1	90.6	2.3	4.4	111	0.162	98.6	14.1
30	8	reg.then.org	45	decentral	4	56.1	73.2	51.4	2.3	4.7	64	0.125	98.5	0.0
31	24	reg.then.org	45	decentral	2	92.9	87.9	94.3	2.3	4.2	123	0.127	98.4	11.6
32	24	reg.then.org	45	decentral	4	65.6	89.1	59.1	2.3	4.5	79	0.115	98.6	0.0

Univariate Analysis

As its name implies this analysis sought to determine the effect on patient time in system as the consequence of varying a single policy or resource constraint. The univariate analysis had two goals. The first was to determine if a change in policy or resource caused a 6 hour difference in the mean patient time in system from the baseline case. The second was to verify that the factors initially thought to be important were actually so. Two statistical tools were used to answer these questions.

Difference of Means. The analyst used the *t* test to compare the mean values of the differing policies. The notation used to describe the test comes from Mendenhall, Wackerly and Scheaffer'. presentation of the topic (26:457-459). Often called the two-sample *t* test, it proves robust to the assumption of normality and to the assumption of equal variances when the samples sizes are equal (as in this case) (26:459). The test takes the form:

$$Ho : u_1 - u_2 = Do$$
$$Ha : u_1 - u_2 > Do$$
$$\textit{Test Statistic}: T = \frac{\bar{Y}_1 - \bar{Y}_2 - Do}{S\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \quad \textit{Rejection Region}: t > t_{\alpha,4}$$

where,

u_1 & u_2 *are two normal populations with equal variances*
$\bar{Y}_1$& $\bar{Y}_2$ *are the sample means*
Do is a fixed value

$$S = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}}$$

S_1^2 & S_2^2 *are the sample variances*
n_1 & n_2 *are the sample sizes*

Again, Table 4.1 provides the results from the 24 sensitivity runs. Each run consisted of 5 replications of the simulation. Run number 2 serves as the baseline. Runs 1 through 7 examine the effects of varying the theater regulation frequency from every 4 hours to every 48 hours. Three input parameters, strategic regulation policy, number of aircraft, and type of command and control are varied in runs 8 through 13. Runs 14 through 20 vary the number of aircraft available. Run 20 also investigates the effect of decentralized command and control with the lowest number of aircraft. The final set of runs, 21-24, look at how changing MOG, the aggregated representation of the APOE resource, affects time in system. The table also reports the values of the other eight output variables.

For each major policy or resource change a *t*-test, at the 5% level of significance, was performed to check whether the difference between the average patient time in system between the baseline and change was more than 6 hours. Table 4.4 summarizes the results. Note that changes in regulation policy, either by decreasing the regulation frequency to every 16 hours or choosing to regulate first to regions, resulted in significant, but opposite, changes to time in system. The choice to regulate first to CONUS regions forced the most dramatic improvement, a nearly 25% reduction in average time in system. In fact, every sensitivity run made that used the regulation policy "region then organization" resulted in a decrease in time in system of the same magnitude than when the "oranization the region" policy was used (see runs 8 versus 9 and 11 versus 12). Increasing the regulation frequency to once every 4 hours for each theater (run 1) was the only other policy change that decreased time in system. Further decreasing regulation frequency (runs 3-7) steadily degraded time in system.

Table 4.4. Summary of Changing AE Policy or Resources

Run	Policy/Resource Change	Mean TIS (hours)	6 hr Difference at 5% Level of Significance?
2	Baseline	73.1	-
4	Theater Regulation - 16 hours	83.5	Yes
10	Regulation Policy - Region First	56.2	Yes
11	Decentralized Command & Control	74.6	No
14	15 Aircraft	75.0	No
20	15 Aircraft, Decentralized Control	116.8	Yes
21	MOG Resource - 2	108.4	Yes
22	MOG Resource - 4	73.8	No

It is interesting that decreasing the number of aircraft from 45 to 15 (run 14) only slightly increased the time in system, as did changing to decentralized command and control (run 11). However, when the combination of these two changes was applied (run 20), average time in system ballooned to 116.8 hours. While time in system was rather insensitive to changes in the number of aircraft, note that (as expected) measures for average and maximum utilization rates steadily climbed as the number of aircraft was decreased, reaching high levels of an average rate of 7.6 and a maximum rate of 15.2 flight hours per aircraft per day when 15 aircraft were in operation (see Table 4.1).

The aggregated APOE resource was insensitive to increases from its baseline value of three. However, when 1 unit of MOG was removed (run 21), time in system rose dramatically to 108.4 hours. These results indicated to the analyst that the aggregated form that APOE resources had been modeled in had introduced a lack of fidelity that requires attention. Suggestions to remedy this situation are given in chapter 5.

Analysis of Variance (ANOVA). While the difference of means test used data from the sensitivity runs, the ANOVA used the 32 design points from the full factorial experiment with time in system designated as the dependent, or response variable. The sole purpose of this analysis was to investigate the magnitude or relative importance of the five main input variables and to check for the existence of interaction between these factors.

The ANOVA was performed using the STATISTIX software package (36:187-215) on the 32 runs from the full factorial 2^5 designed experiment (see Table 4.3; all the output variables were defined earlier in this chapter) with average time in system as the dependent variable. The resulting ANOVA table shown at Table 4.5 annotates significant effects and interactions at the 5% level of significance with a double arrow. Only significant three-way and higher interactions are listed.

Table 4.5. ANOVA Table for Average Time in System

Source	DF	Sum of Squares	Mean Squares Error	F	P-value	
Reg Frequency (A)	1	508.0	508.0	33176.0	0.0035	<<
Reg Policy (B)	1	5379.4	5379.4	351310.2	0.0011	<<
Number A/C (C)	1	2392.5	2392.5	156250.8	0.0016	<<
Cmd & Control (D)	1	1408.5	1408.5	91982.2	0.0021	<<
MOG (E)	1	8827.9	8827.9	576514.8	0.0008	<<
A*B	1	2.4	2.4	154.5	0.0511	
A*C	1	7.5	7.5	490.3	0.0287	<<
A*D	1	1.6	1.6	102.8	0.0626	
A*E	1	40.3	40.3	26030.2	0.0124	<<
B*C	1	257.1	257.1	16788.8	0.0049	<<
B*D	1	215.8	215.8	14093.1	0.0054	<<
B*E	1	33.4	33.4	2182.2	0.0136	<<
C*D	1	1459.4	1459.4	95304.5	0.0021	<<
C*E	1	73.5	73.5	4800.5	0.0092	<<
D*E	1	0.4	0.4	25.0	0.1257	
A*B*E	1	5.0	5.0	329.2	0.0351	<<
B*C*D	1	242.6	242.6	15840.0	0.0051	<<
B*C*E	1	5.5	5.5	361.0	0.0335	<<
B*C*D*E	1	3.9	3.9	251.5	0.0401	<<

All main effects and all but two of the two-way interactions are significant. Among main effects, the MOG resource seems most influential followed by the regulation policy and number of aircraft. This confirms our experts' intuition of what factors are important. The fact the MOG resource is most influential should not be surprising. After all, the APOE defines the interface between the medical system and airlift system. The resources available at the APOEs will influence operations that both feed and retrieve patients from these locations. There is also significant interaction at the two-factor level and even some at the three-factor level, highlighting the fact that AE is a complicated business, but also one that possesses many tradeoffs, as is shown in the multivariate analysis.

Multivariate Analysis

Unlike the univariate techniques mentioned above, multivariate techniques seek to unveil the simultaneous relationship among a collection of multiple output variables (nine have been recorded in the scenario output). Dillon and Goldstein, in their text (14), define multivariate analysis as "the application of methods that deal with reasonably large numbers of measurements (i.e., variables) made on each object in one or more samples simultaneously" (Here, the term "object" refers to a run of the simulation model) (14:1). They go on to say that this type of analysis differs from univariate and bivariate analyses in that it directs attention to the correlation amongst the multiple (three or more) variables (14:2). Two of the methods they describe have application to analysis of simulation output. These techniques are known as principle component analysis (PCA) and factor analysis.

The primary purpose of using principal component analysis and factor analysis is to better understand this "relationship" that exists among the strategic AE simulation output in hopes that it will deliver insights to policies and resources under the AMC medical planner's control.

To gain relational insights about the strategic AE process the analyst performed both a principal component and factor analysis on the output data from the designed experiment (see Table 4.3). After initial examination of this data it was decided to drop two of the nine output variables before proceeding with the analysis. The variables measuring percent of patients transferred to the CONUS and average aircraft parked at 4E facilities were dropped because they showed very little sensitivity to the input parameters. The near constant percentage of patients delivered can be attributed to the demand-driven logic.

For this study, the analyst used principal components analysis to identify factors that explained most of the variance of the output vector. With this initial estimate of what and how many factors were important, the analyst then performed a factor analysis, plotting factor scores in search of relationships between the factors and original simulation input variables.

Principal Components Analysis. The overall objective of PCA is to study the interdependence structure of a set of variables. PCA is a useful data reduction technique that seeks to find the true dimensionality (number of major drivers) and an interpretation for the data. The basic premise is that the elements of the output vector of the simulation are interrelated and that "these variables are really measuring some underlying or latent factors" (2:15). The goal in PCA is to form a linear combination of the original output vector that accounts for most of the total variation in the output variables (14:53).

As a data reduction technique the idea behind this type of analysis is "to transform the original set of variables into a smaller set of linear combinations that accounts for most of the variance of the original set" (14:24). To extract the principal components, usually the data is transformed to either a covariance or correlation matrix. Normally, if the units and scales for the data are different, as was in this analysis, the correlation matrix is used (14:26).

Conveniently, it results that the first principal component is associated with the largest normalized eigenvalue from the matrix, the second principle component with the next largest eigenvalue, etc. The total variance is defined by the sum of the eigenvalues. The amount of total variance explained by each principal component is simply the value of its associated eigenvalue divided by the sum of the eigenvalues for the matrix.

The component loadings, how each variable loads on the principal component, are used to help interpret what the principal components represent (11:31). Usually, after the number of principal components to keep for interpretation has been decided (normally when most of the variance is explained), each variable's highest loading is identified. The analyst then attempts to assign a meaning or interpretation to the set of loadings for each principal component.

The SAS principal component procedure was used to perform the analysis. This procedure is explained in the SAS Procedures Guide (31:751-771). The SAS run yielded the following eigenvalues from the correlation matrix, their relative magnitude compared with other eigenvalues, and the amount of variance explained by each (reference Table 4.6). Because the output variables are in different units, the correlation matrix was used for the analysis.

Table 4.6. PCA, Eigenvalues from the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative	
PRIN1	4.36696	2.55774	0.623851	0.62385	
PRIN2	1.80921	1.08159	0.258459	0.88231	
PRIN3	0.72762	0.63737	0.103946	0.98626	<<<
PRIN4	0.09025	0.08514	0.012893	0.99915	
PRIN5	0.00511	0.00427	0.000730	0.99988	
PRIN6	0.00084	0.00084	0.000121	1.00000	
PRIN7	0.00001	0.00000	0.000001	1.00000	

Even though the idea of PCA is to reduce the original number of variables to a smaller set of linear combinations that explain most of the variance, the analyst decided to keep the first three principal components for interpretation. Most rules (such as the scree test and eigenvalues greater than 1.0) mentioned by Dillon and Goldstein (14:47-49) would suggest keeping only the first two principal components for interpretation. However, since the third principal component does account for more than 10% of the total variance, it was kept, and thus 98.6 % of the total variance is explained in the first three principal components. The next step was to determine what these components mean or may represent in terms of the strategic AE scenario they reflect.

Recall that component loadings, or how much each variable "loads on each component" can be found by extracting the eigenvectors from their associated eigenvalues. Table 4.7 provides the eigenvectors for the first three principal components.

Table 4.7. PCA, Eigenvectors

	PRIN1	PRIN2	PRIN3
Time in System (TIS)	0.436193	-.290825	-.131236
TIS Southwest Asia	0.227899	-.377641	0.837015
TIS Far East	0.430551	-.235548	-.338956
Avg Ute Rate	0.338194	0.509565	0.193805
Max Ute Rate	0.302062	0.547780	0.258079
Avg # in 3E Hospitals	0.434848	-.302824	-.066548
% Missions Delayed	0.421561	0.256716	-.242276

For each loading the highest absolute loading has been underlined. Interpretation will be based on the group of variables loading on each component. It appears the first component is a good overall measure of patient handling since the first eigenvector shows almost equal loadings on all the variables, but particularly those measuring patient time attributes.

The second principal component shows heavy loadings on the two measures of aircraft use, average utilization rate and maximum utilization rate, with negative loadings on all the other variables except percent missions delayed. The signs make sense, in general, given greater aircraft utilization, the patient time in system measures and number of patients in 3E hospitals decrease, and the percentage of missions delayed increases. (Remember, for low numbers of aircraft, utilization per aircraft increased but more missions can be delayed.)

The third principal component is loaded on heavily by a single variable, time in system for the Southwest Asia theater. This points out a peculiar phenomenon associated with this two-theater scenario (remember the warning at the beginning of this chapter).

Note the opposite signs on the other two time in system measures. Apparently time in system for Southwest Asia increases at the expense of lower times in the Far East. Remember that mission flight times are shorter for the Far East and more aircraft are made available to the Far East with a decentralized command and control policy. In the last half of the simulation most patients are predominantly coming from the Far East at a much larger rate than Southwest Asia. Consequently, the Far East theater is dominating the use of the aircraft resource, particularly with a decentralized policy. Compare runs 2 & 18 and 4 & 22 of the designed experiment for an example of this.

The PCA thus identifies two clear factors or principal components, one associated with patient attributes and the other aircraft usage which explain most of the variance. The third principal component is a little fuzzier in terms of interpretation. Often, factor analysis will yield similar results with better interpretation.

Factor Analysis. Factor analysis is another interdependence technique that is oriented toward common variation among the variables versus PCA's orientation toward total variation (2:42). Dillon and Goldstein define factor analysis as,

> ...the study of interrelationships among the variables in an effort to find a new set of variables, fewer in number than the original set of variables, which express that which is common among the original variables... (14:53)

They summarize the purpose of factor analysis well when they state that it "attempts to simplify complex and diverse relationships that exist among a set of observed variables by uncovering common dimensions or factors that link together the seemingly unrelated variables, and consequently provides insight to the underlying structure of the data"

(14:53). There are two major applications of factor analysis, exploratory, where a search for common structure to the data is the goal, and confirmatory, where a test is made on some prior hypothesis (2:43). For this application the analyst applied the technique in the exploratory sense.

The basic common factor model takes the form (14:61)

$$X = \Lambda f + e$$

where,

X = p-dimensional vector of observed responses
f = q-dimensional vector of unobservable variables called common factors
e = p-dimensional vector of unobservable variables called unique factors, and
Λ = p x q matrix of unknown constants called factor loadings,

$$\Lambda = \begin{pmatrix} \lambda_{11} & \lambda_{12} \dots & \lambda_{1q} \\ \lambda_{21} & \lambda_{22} \dots & \lambda_{1q} \\ . & . & . \\ . & . & . \\ \lambda_{p1} & \lambda_{p2} \dots & \lambda_{pq} \end{pmatrix}$$

The loadings, Λij, provide the correlation between a variable and a factor and in a sense relate the degree each variable loads on a factor. Often, interpretation of these loadings is confusing. Dillon and Goldstein offer a procedure to simplify the process. This procedure was used by the analyst and guides the discussion of factor analysis results (14:69).

Sometimes rotation of the factors can help simplify the structure and improve interpretation. The most common method, and the one used in this analysis is called varimax rotation, which attempts to maximize variation of the squared factor loadings within a factor (2:58-59).

Finally, it is extremely useful to estimate the factor scores and then plot them. By annotating the plot with the values of the original input variables of the simulation some interesting relationships begin to surface. Factor scores provide "the location of each observation in the space of common factors" (14:96). Unlike principal component scores, which can be calculated directly as linear combinations of the original variables, factor scores must be estimated, usually by means of multiple regression analysis (14:96). For this analysis, the three factors were not plotted in three space, but plotted two at a time to aid in interpretation.

Based on the results of the principal component analysis, the analyst decided to run the SAS factor procedure specifying the number of factors as 3. A complete explanation of its use is found in the SAS Procedures Guide (31:449-492). The SAS procedure uses the principal component factor analysis method and produces the following table:

Table 4.8. Initial Factor Method, Principal Components

Eigenvalues of the Correlation Matrix: Total = 7 Average = 1

	1	2	3	4	5	6	7
Eigenvalue	4.3670	0.8092	0.7276	0.0903	0.0051	0.0008	0.0000
Difference	2.5577	1.0816	0.6374	0.0851	0.0043	0.0008	0.0000
Proportion	0.6239	0.2585	0.1039	0.0129	0.0007	0.0001	0.0000
Cumulative	0.6239	0.8823	0.9863	0.9991	0.9999	1.0000	1.0000

Again, it appears that the first three factors explain most of the variance. The resulting factor pattern (which shows the correlation between each variable and the unobserved factor) is shown in Table 4.9.

Table 4.9. Factor Pattern

	FACTOR1	FACTOR2	FACTOR3
Time in System (TIS)	0.91152	-0.39118	-0.11195
TIS Southwest Asia	0.47625	-0.50795	0.71398
TIS Far East	0.89973	-0.31683	-0.28913
Avg Ute Rate	0.70673	0.68540	0.16532
Max Ute Rate	0.63123	0.73680	0.22014
Avg # in 3E Hospitals	0.90871	-0.40732	-0.05677
% Missions Delayed	0.88095	0.34530	-0.20666

Variance explained by each factor

FACTOR1	FACTOR2	FACTOR3
4.366956	1.809213	0.727619
(63%)	(26%)	(11%)

Following the procedures outlined by Dillon and Goldstein (14:69) an attempt was made to interpret the 3 factors. As with the principal components, the loading that contributed most to each variable was underlined. A judgement was then made to the statistical significance of each loading. Normally with sample sizes of less than 100, the loading's absolute value needs to be greater than 0.30 to be considered significant (14:69). This was the case with all the loadings above. Note also the proportion of total variance explained by each factor. Factor 1 explains approximately 63.3% of the total variance captured by the three factors, factor 2 explains approximately 26.2%, and factor three accounts for 10.5%.

Having several variables with moderate size loadings often complicates interpretation (14:69). Therefore, varimax rotation, discussed earlier, was applied with the goal of minimizing the number of significant loadings for each variable. Table 4.10 shows the new rotated factor pattern.

Table 4.10. Rotated Factor Pattern Using Varimax Method

	FACTOR1	FACTOR2	FACTOR3
Time in System (TIS)	0.94147	0.17157	0.28393
TIS Southwest Asia	0.29847	0.00998	0.95153
TIS Far East	0.97443	0.18704	0.09489
Avg Ute Rate	0.18852	0.97940	0.04248
Max Ute Rate	0.08291	0.99009	0.05140
Avg # in 3E Hospitals	0.92299	0.16875	0.33837
% Missions Delayed	0.63350	0.72505	-.10480

Variance explained by each factor

FACTOR1	FACTOR2	FACTOR3
3.220623 (47%)	2.558205 (37%)	1.124961 (16%)

These results are very similar to the results observed in the PCA, which is often the case with the two techniques.

To help better understand what the factors mean one can estimate the factor scores and plot them. Figures 4.1-4.5 show the results of plotting the standardized scores for each observation. Table 4.11 contains the standardized scoring coefficients for each of the 32 observations in the designed experiment.

Table 4.11. Standardized Scoring Coefficients Estimated by Regression

	FACTOR1	FACTOR2	FACTOR3
Time in System (TIS)	0.32554	-0.08946	0.00328
TIS Southwest Asia	-0.20543	0.04381	1.00472
TIS Far East	0.40838	-0.10965	-0.22851
Avg Ute Rate	-0.13453	0.44238	0.08680
Max Ute Rate	-0.19261	0.47149	0.13743
Avg # in 3E Hospitals	0.29686	-0.08090	0.07353
% Missions Delayed	0.19609	0.20549	-0.27770

By studying Figures 4.1-4.5 in conjunction with Table 4.4, specifically the input parameters associated with each observation, one can assign a label to each of the factors and begin to better understand which input parameters influence each factor.

Figure 4.1, the plot of factor 1 versus factor 2, revealed several things. First note the two distinct groups of data on the factor 2 axis. Clearly, the number of aircraft determines the sign of factor 2, which the analyst thus labeled Airlift Resources. It is also interesting that several items influence variance along the axis of factor 1. The primary variable is MOG, with higher values of MOG being toward the bottom of the graph. For this reason the factor was labeled APOE Resource. Within the MOG subgroup, another set of groups is defined by the strategic regulation policy, with "region then organization" producing lower factor 1 scores. Note, in general the lower the factor 1 score, the lower the overall time in system and time in system to the Far East since they have such heavy loadings on factor 1 (remember that lower time in systems are desirable).

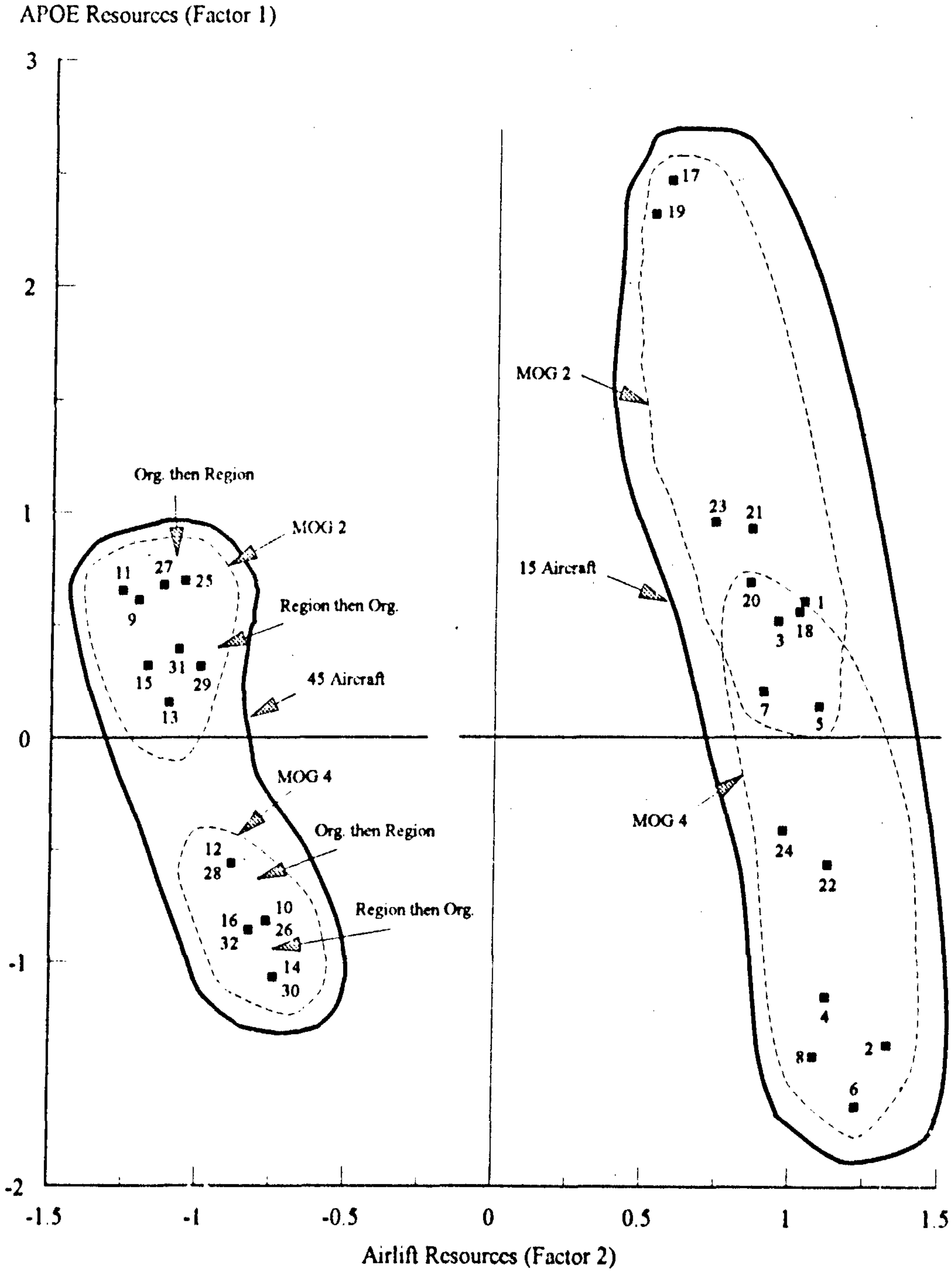

Figure 4.1. Plot of Factor 1 vs Factor 2

Figure 4.2, the plot of factor 1 versus factor 3, again reveals the importance of the APOE resource on factor 1 observation scores. Note the two large groupings of factor scores. In general, observations with positive factor scores had a MOG value of 2, those with negative scores had a high MOG value of 4. The exceptions were observations 18 & 20. These two points were pushed up by the fact that there were a low number of aircraft used and there was decentralized command and control, an event also observed in the sensitivity analysis. The variance in factor 2 is clearly attributed to the strategic regulation policy. After looking at Figure 4.3 the discovery is made that the variance within the strategic regulation policy along the factor 2 axis is due to regulation frequency. Therefore, factor 2 is labeled Regulation Policy/Coordination. Note the tight variance about the "region then organization" observations and the wide variance among the "organization then region" observations. This is because the former policy is more flexible in handling lower number of airlift resources and the decentralized command and control.

In Figure 4.3 notice the region in the lower left hand corner containing nine observations where the average number of occupied beds in 3E facilities is less than 90. This is one of several tradeoffs that can be unveiled in these types of graphs. By committing resources to the APOE and selecting the right strategic regulation policy, resources required at the 3E facility could be reduced. Note also that, in general, this same area of the plot has the observations with the lowest time in system measurements for the patient.

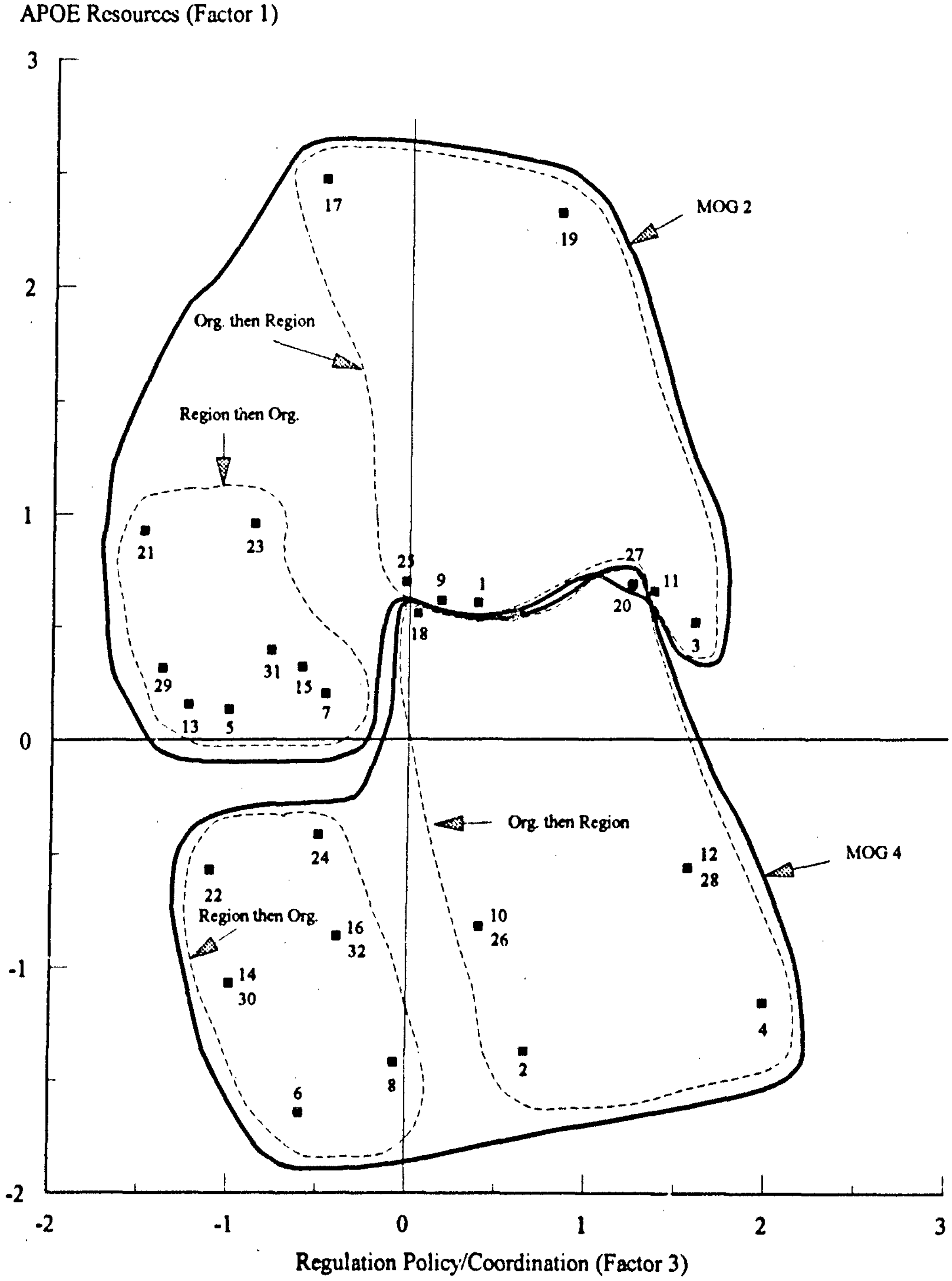

Figure 4.2. Plot 1 of Factor 1 vs Factor 3

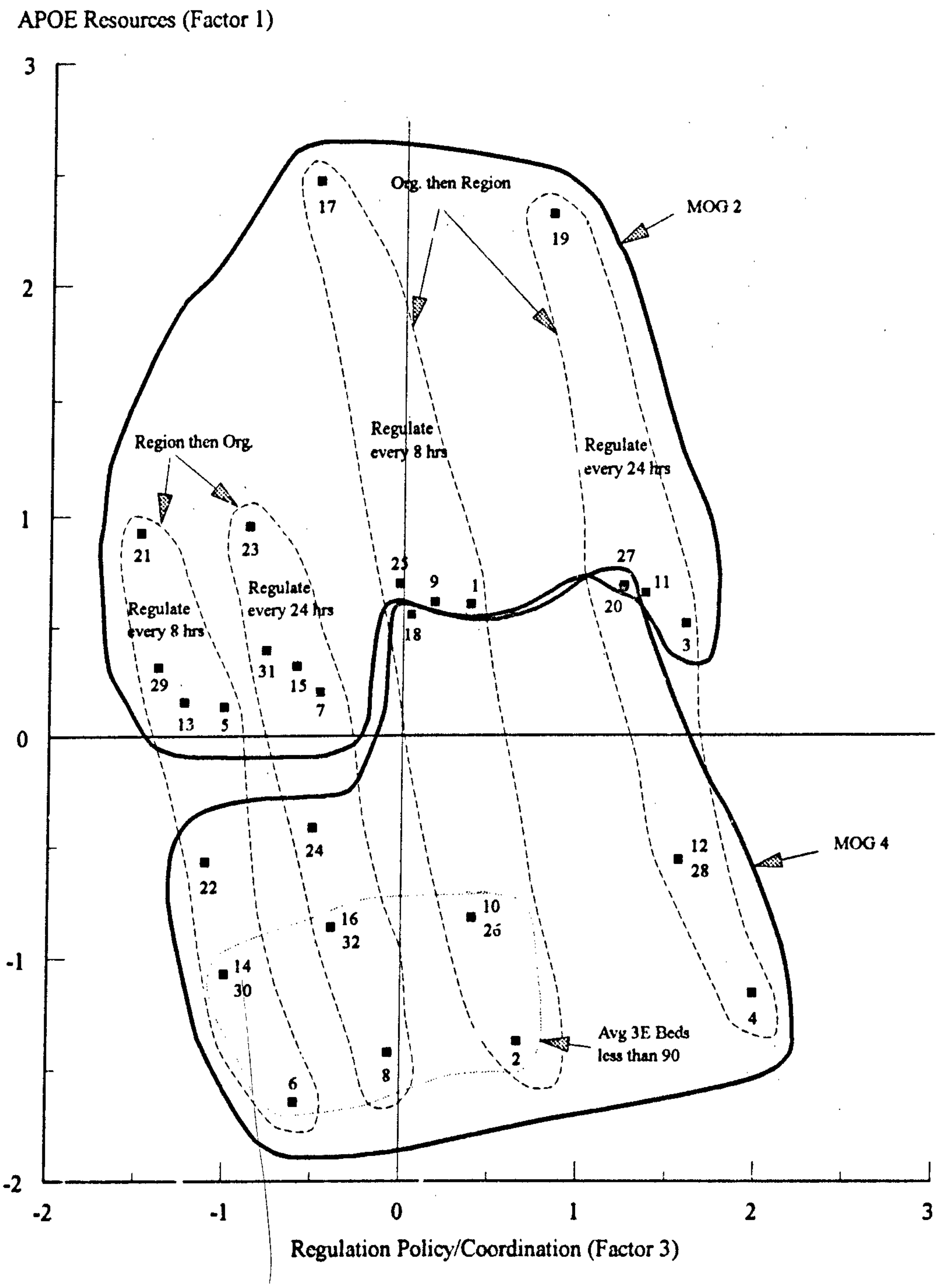

Figure 4.3. Plot 2 of Factor 1 vs Factor 3

Figure 4.4, the plot of factor 2 versus factor 3, again showed a big split in the observations along the Airlift Resources factor axis. Note the higher the number of aircraft the lower the factor score. Remember that utilization rates load heavily on this factor. The higher the ute rates, the higher the score will be. A low number of aircraft results in higher ute rates, and thus higher factor scores.

Again, the variance along the Regulation Policy/Coordination axis is defined by the strategic regulation policy and the regulation frequency. Remember that factor 3 was heavily loaded by time in system for the Southwest Asia theater. As time in system for the Southwest Asia theater decreased, so did the factor scores. Note that observations on the left side of the graph provide a more balanced time in system between the theaters of operation. This would more than likely be a goal given the theaters were producing the same type of casualties. Finally, with 15 aircraft, the tradeoffs between regulation policy and frequency become more convoluted, whereas with a large number of aircraft, options for tradeoffs are more clear. It makes sense that the more resources one has the more options there should be.

Figure 4.5 is the same as Figure 4.4 only the lower half of the plot shows the region where no mission delays occurred. Essentially, all the observations with 45 aircraft experienced no mission delays with the exception of observations where there was limited MOG and decentralized control of strategic aircraft. No mission delays are not necessarily good, since that means aircraft were idle a great deal of the time.

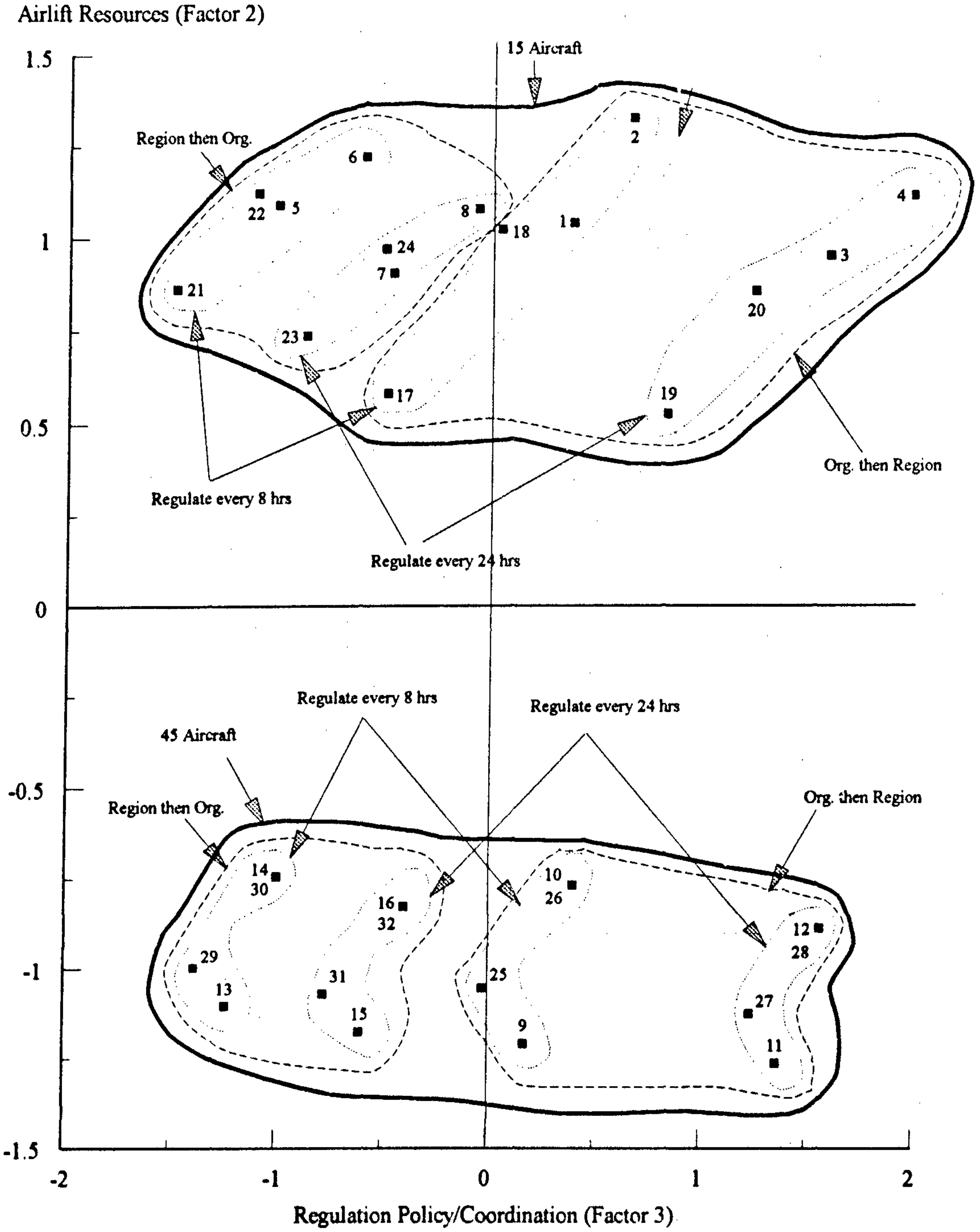

Figure 4.4. Plot 1 of Factor 2 vs Factor 3

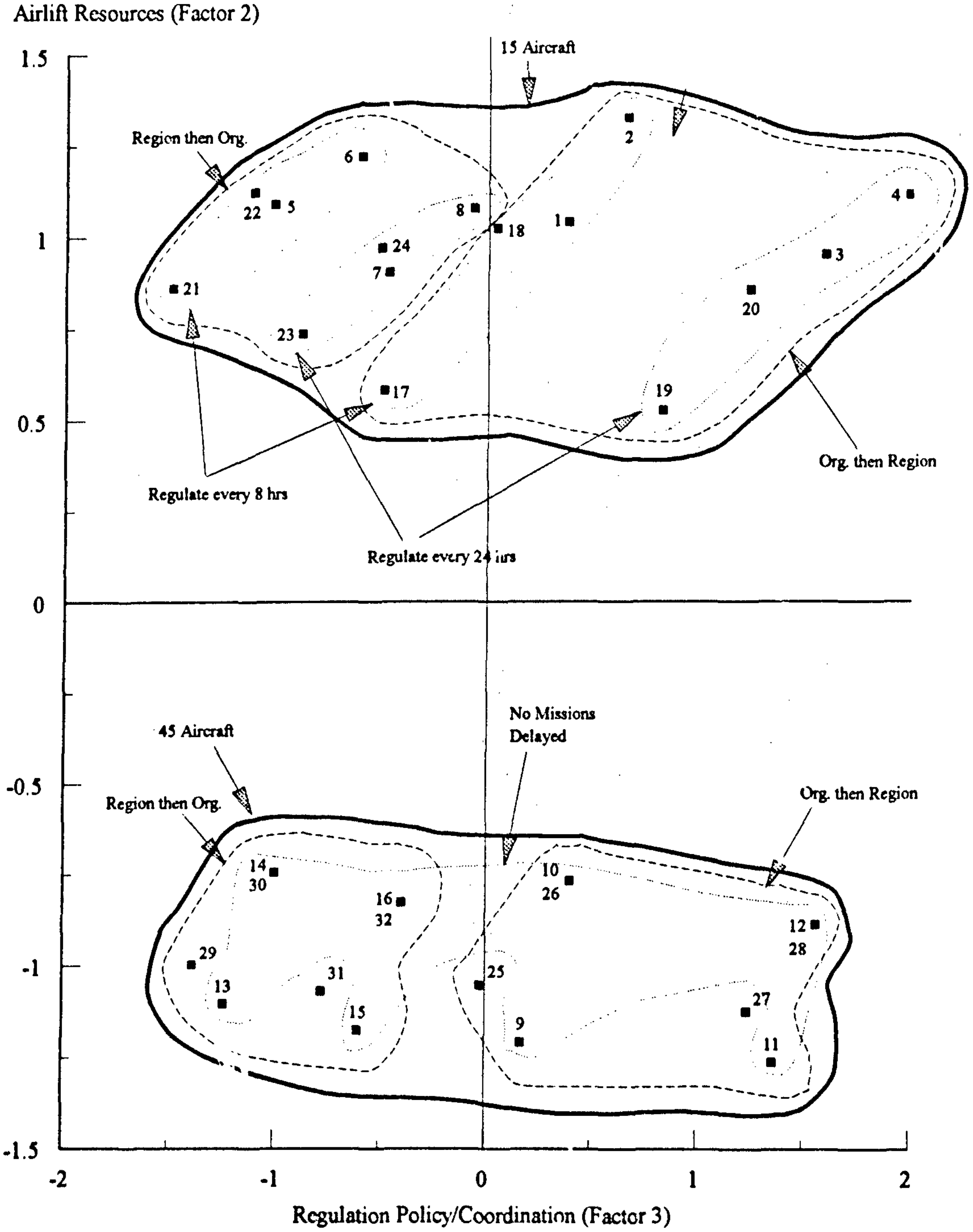

Figure 4.5. Plot 2 of Factor 2 vs Factor 3

Many, many more inferences can be made from these plots. The poini :s, they unveil what the main factors are and how they are related, and they spur the planner to ask key what-if questions which can easily be answered by running the simulation again.

Principal components pointed the a.ıalyst toward the two general areas of measurements on the patient and aircraft usage, and a third component that loaded on time in sytem for the Southwest Asia theater, which was unclear in definition. PCA also provided a direction for the factor analysis. After using the varimax rotation, the factor analysis showed similar loadings to the principal component analysis. But after plotting the factor scores and overlaying the simulation input variables, the meaning of the factors became cleer, as well as the vast potential for tradeoffs depending on the end user's objectives.

V. Conclusions & Recommendations

This chapter summarizes some general conclusions and provides recommendations for the model's fidelity and expansion, possible uses of the model, and additional research.

Insights

Three main factors significantly affect strategic AE operations. They include the resources located at an APOE, the regulation policies defined by the ASMRO, and the amount of strategic aircraft available. For the defined scenario, changing the regulation policy to fill CONUS regions first rather than organizations first (i.e., all the DOD hospitals, then all the VA, then all the NDMS) reduces average time in system for a patient by approximately 25 percent. The analyses identified that a great deal of interaction exists between the major elements of strategic AE. Combining decentralized command and control and a low number of strategic aircraft was consistently detrimental to average time in system for the patient. The analyses also showed that there is a vast potential for tradeoffs, depending on the end user's objectives.

A Flexible Planning Tool

The objectives of this research outline some desirable characteristics that the simulation model should possess in order to better serve the medical planning community. Specifically, the model needs to incorporate the major elements of strategic AE, be modular in nature to facilitate maintenance and future enhancements, and have the capability to quickly answer what-if type questions. The model meets all these requirements.

Chapter 3 highlighted how the simulation model captures the major elements of strategic AE through the code's modules which were based on major AE processes. Some of these include the echelon care system, patient regulation, APOE resources, and mission planning and execution.

Scenario changes are easily handled through changing the data input file. There is no need to recode the simulation in order to answer many of the common what-if type questions which will arise. For example, the provided scenario was translated into an input file for the simulation in a matter of a day. Changes such as adding additional APOEs, aircraft, or even another theater of operation can be accomplished in a matter of minutes. While the substance of the code forms an adequate baseline for an initial study, there were areas identified to improve the model's fidelity and ease of use.

Model Fidelity & Expansion

During the course of building the simulation it became apparent that the code could be improved to better represent certain elements of strategic AE. Specifically, the sensitivity analysis pointed out that the MOG resource, which represents several APOE resources, should be decomposed and modeled explicitly.

There are several reasons MOG should be modeled explicitly. First, both analyses, particularly the multivariate, showed the importance of APOE operations and its effects on strategic AE performance measures. As previously mentioned, this should come as no surprise since the APOE is the area where the medical care system interfaces with the strategic airlift system. Second, the model does not currently consider how well strategic aircraft will be able to cycle medically qualified aircrews between the theaters of operations and the APOEs. It just assumes the aircrew resources are there when an aircraft seizes the MOG resource. The ability to better control the return of aircrews to the APOE was one reason the aeromedical community originally sought a dedicated strategic aircraft. This model should have the capability to track the use of the aircrew resource and evaluate how this resource affects total strategic AE operations. Finally, the model needs to explicitly represent ramp space and fueling operations at the APOE airfield, since it is likely these APOEs will be collocated at airfields where tanker, cargo, or even tactical aircraft may reside.

Another area that the model did not incorporate in this study but which requires attention is maintenance of the Boeing 767 fleet. This should include both scheduled and unscheduled maintenance and the locations where each type of maintenance may be performed.

The scope of this research was limited to strategic AE. Now that baseline code has been written, a next step might be to expand this code to include tactical movement of casualties in the theater of operations and CONUS redistribution of patients to their final care destinations. Then AMC/XPY could perform studies and examine tradeoffs between the three major AE operations: intertheater, intratheater, and domestic. One particular question this type of study could address is the benefit of flying patients directly to their end care facility rather than just the CONUS hub (reference AFIT Thesis, *Patient*

Scheduling & Aircraft Routing for Strategic Aeromedical Evacuation) (24). The modular design of the code written for this research will ease the effort required to expand the simulation in order to help answer these types of questions.

Uses

There are many possible uses for this simulation model. While the analyses in this research focused on patient time in system, there are many other output measures that medical planners are obviously interested in. Here are a few examples.

The simulation could be used to assist CRAF activation planning to even include estimating the cost of different activation options based on an expected utilization rate of the fleet. The analyst can quickly look at different fleet configurations, not only considering varying number of aircraft, but also different aircraft patient capacities. For example, for the scenario provided in this research, the analyst could examine the influence of assigning a higher patient capacity aircraft to the Southwest Asia (longer) routes.

Policy surrounding patient regulation was found to heavily influence patient time in system in this research. The relative effect that regulation policy will have will depend upon the scenario conditions. AMC analysts could work with medical regulators to identify the set of regulating policies which will work best under the most common scenario conditions. Another related area of interest is the filling of CONUS beds For different scenarios regulators could identify and plan for bed shortages by patient type. Also the effect of limiting bed availability to certain organizational types, such as just DOD, or DOD and VA only could be studied.

The simulation could also be used to study broad medical resource allocation tradeoffs. For example, for the two theater scenario studied in this research, certain policy/resource combinations resulted in a lower average number of patients in all 3E facilities. That combination may be characterized by an increase in the number of CRAF aircraft. Medical planners could use the simulation to study the training and skill requirements associated with allocating medical personnel to aircrews or to manning 3E type facilities.

Another use for the simulation was alluded to in the research objective for this thesis. That is, the simulation provides AMC/XPY with a generic stochastic tool to verify some of the resource sizing recommendations they make using deterministic tools.

There are many more topics that could be discussed, but the point is that the model has the fidelity and flexibility to address these types of questions fairly quickly. After the model is used to answer some of these type questions, no doubt it will spur the medical planners to explore even more options and ask more questions. The proper application of this tool will in the end result in better medical contingency plans.

Additional Research

The next step in research in this area lies in exploring the effects of different policy/resource recommendations across a suite of different scenarios to identify the best matches. To facilitate this an automated scenario generator could be built to ease and speed up the front-end work required of the analyst to prepare for a simulation run.

Bibliography

1. Alfano, Joseph P. and O'Neill, John C., *Wartime CONUS Casualty Distribution System Using Dedicated CRAF Airlift*, MS Thesis, AFIT/GST/85M-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1985.

2. Bauer, Kenneth W., Class Notes, Oper 685, Applied Multivariate Analysis, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, Fall 1992.

3. Berger, Carl, *The United States Air Force in Southeast Asia, 1961-1973*, U.S. Government Printing Office, Washington D.C., 1977.

4. Booze, Allen & Hamilton. *External Logistics Processor, LPX Medical Module, User/System Reference Manual (Draft)*, March 1992.

5. Bossart, Bruce, Armed Services Medical Regulating Office, Scott AFB, IL, Telephone Interview, January 1993.

6. Burnes, Michael D., *Application of Vehicle Routing Heuristics to an Aeromedical Airlift Problem*, MS Thesis, AFIT/GST/90M-3. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1990.

7. Carter, William B., *Allocation and Routing of CRAF MD80 Aircraft*, MS Thesis, AFIT/GST/90M-4. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1990.

8. Clingman, Fred M., *Analysis of Aeromedical Evacuation in the Korean War and Vietnam War*, MS Thesis, AFIT/GLM/LS/89S-9. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, September 1989.

9. Department of Defense, Secretary of Defense, *Policy with Respect to Evacuation of Patients*, 7 September 1949.

10. Department of the Air Force, Concept Paper, *Aeromedical Evacuation - The Future*, Headquarters Air Mobility Command, Surgeon General, Scott AFB, IL, June 1992.

11. Department of the Air Force, *Contingency Aeromedical Evacuation - A Primer*, 1st Aeromedical Evacuation Squadron, Pope AFB, NC, 1 May 1992.

12. Department of the Air Force, Department of the Surgeon General, *A Concise History of the USAF Aeromedical Evacuation System*, U.S. Government Printing Office, Washington D.C., 1976.

13. Department of the Air Force, *MAC Aeromedical Evacuation (AE) Concept of Operations Phase II.* Prepared by Analytical Systems Engineering Corporation. Contract F19628-86-D-0003, 15 June 1989.

14. Dillon, William R., and Goldstein, Matthew, *Multivariate Analysis, Methods & Applications*, New York: John Wiley & Sons, 1984.

15. Dobbs, Rebecca A., and others, *Concept Development for an Aeromedical Data Acquisition and Communications System (AMDACS)*, Interim Report, Battelle Columbus Division, Columbus, OH, August 1990.

16. Ewing, William B., *Casualty Evacuation and Distribution Using B-767 and C-9 Aircraft*, MS Thesis, AFIT/GST/OS/86M-8. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1986.

17. Funsch, Harold F., Nareff, Max J., and Watkins, Philip B., *Wings for Wounded Warriors*, Journal of the American Medical Association, May 1967.

18. Futrell, Robert F., *Development of Aeromedical Evacuation in the USAF, 1909-1960*, USAF Historical Study 23, USAF Historical Research Center, Maxwell AFB, AL, 1960.

19. Johnson, Kathleen, *The Aeromedical Evacuation of Non-DOD Civilians*, Research Report, Industrial College of the Armed Forces, Ft McNair, Washington D.C., 1987.

20. Law, Averill M. and Kelton, David W., *Simulation Modeling and Analysis*, McGraw-Hill Book Company, 1991.

21. Lee, Jerry N., *Toward a Healthier Aeromedical Operation: Integrating Medical Policies and Airlift Concepts*, Research Report, Air University Press, Maxwell AFB, AL, July 1986.

22. Litko, Joseph, Air Mobility Command Analysis Group, Air Mobility Command (AMC), Scott AFB IL. Personal Interview, October 1992.

23. Llewellyn, Craig H., and others, *Wartime Strategic/Domestic Aeromedical Evacuation and Distribution of Patients*, Research Report, Industrial College of the Armed Forces, National Defense University, Ft McNair, Washington D.C., 1982.

24. Loftus, Michael J., *Patient Scheduling and Aircraft Routing for Strategic Aeromedical Evacuation*, MS Thesis, AFIT/GST/93M-7. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1993.

25. Mahlem, Phil, Air Mobility Command Surgeon's Staff, Air Mobility Command, Scott AFB, IL. Telephone Interview, February 1993.

26. Mendenhall, William, Wackerly, Dennis D., and Scheaffer, Richard L., *Mathematical Statistics with Applications, Fourth Edition*, Boston: PWS-KENT Publishing Company, 1990.

27. Nagel, Susan K., and others, *Evaluation of a Proposed North Atlantic Treaty Organization (NATO) Standard Litter*, Research Report, Armstrong Laboratory, Brooks AFB, TX, March 1992.

28. Negrelli, Edward P., Class Lecture, OPER 775, Combat Modeling (High Resolution), Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December, 1992.

29. Pletcher, Kenneth E., *Aeromedical Evacuation in Southeast Asia*, Air University Review, March-April 1966.

30. Ritchie, Gilbert, and others, *Development & Testing of a Low-Flow Servo Controlled Oxygen Delivery System*, Final Report, Contract F33615-87-C-0623, University of Alabama, Birmingham, AL, March 1990.

31. SAS Institute Inc., *SAS Procedures Guide, Release 6.03,* SAS Institute, Cary, NC, August 1991.

32. Schwartz, Rudy, *Aeromedical Evacuation Contingency Planning*, Research Paper, Naval War College, 19 June 1992.

33. Shrum, James K., *Qualification Operational Test & Evaluation of the Infusion Pump*, Final Report, HQ MAC, Scott AFB, IL, May 1991.

34. Shrum, James K., *Qualification Cperational Test & Evaluation of the Pulse Oximeter*, Final Report, HQ MAC, Scott AFB, IL, May 1991.

35. Smith, Clarence M., *The Medical Department: Hospitalization and Evacuation, Zone of Interior*, U.S. Government Printing Office, Washington D.C., 1956.

36. Statistix Analytical Software, *Statistix User's Manual, Version 4.0,* Analytical Software, St Paul, MN, 1985.

37. Ware, Keith, Air Mobility Command Analysis Group, Air Mobility Command (AMC), Scott AFB IL. Telephone Interview, May 1992.

38. Waters, W. J., *Operational Test & Evaluation of the Two-Wheel Gurney*, HQ MAC, Scott AFB, IL, August 1985.

39. Whetstone, Tod W., *A Heuristic Approach for Aeromedical Evacuation System Scheduling and Routing*, MS Thesis, AFIT/GSO/ENS/88D-20. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1988.

40. White, M.S., *Air Transportation for the Sick and Wounded, USAF Historical Research Center*, Maxwell AFB, AL, 1971.

Appendix A. SIMSCRIPT II.5 Computer Code

```
PREAMBLE

normally, mode is undefined

processes include fly.mission,
                  move.patients.to.4e
                 and stop.simulation
   every fly.mission has a server,
                         a server2,
                         a server3,
                         a server4,
                         a server5,
                         and a server6
   every move.patients.to.4e has a server,
                                 a server2,
                                 a server3,
                                 and a server4

resources include mog,
                  mog.return

events include make.patient,
          regulate,
          check.demand.for.strat.ae,
          mission.generator,
          check.missions.delayed,
          update.parameters,
          and heal
   every make.patient has a server,
                  and a server2
   every regulate has a server,
                 and a server2
   every check.demand.for.strat.ae has a server,
                          and a server2
   every mission.generator has a server,
                     a server2,
                     a server3,
                     and a server4

   every check.missions.delayed has a server
   every update.parameters has a server
   every heal has a server,
            and a server2
```

```
define server,
       server2,
       server3,
       server4,
       server5,
       and server6 as integer variables

permanent entities include route,
                           location,
                           and aircraft

  every route has a route.no,
            a route.name,
            a region.destination,
            a route.theater.no,
            a base.conus.location,
            a no.aircraft.in.route.pool,
            and a route.flight.time,
            and owns a route.leg.sequence,
                   and a route.aircraft.pool

  every location has a location.no,
              a location.name,
              a mission,
               a mean.batch.interarrival.time,
              a min.batch.size,
              a max.batch.size,
              a no.facility.3e.feeders,
              a no.mog,
              a no.mog.return,
              a theater.no,
              a no.planes.parked,
              a mog.in.use,
              a waiting.mog,
              a patient.type.mix random step variable
              and owns a patient.list,
                     and a location.feeder.pool

 every aircraft has a aircraft.no,
             a start.location,
             a present.location,
             a capacity,
             a status,
             a type,
                a in.use,
                a total.on.board,
                a ac.flight.time,
                a int.ac.flight.time,
                a no.missions.flown,
                a total.ac.flight.hours,
                a int.total.ac.flight.hours,
                and owns a manifest.list

define manifest.list as a fifo set
define route.leg.sequence as a fifo set
define route.aircraft.pool as a fifo set
define patient.list as a fifo set
define location.feeder.pool as a fifo set
define route.no,
    base.conus.location,
    region.destination,
    route.theater.no,
    no aircraft.in.route.pool,
    location.no,
    no.facility.3e.feeders,
    no.mog,
    no.mog.return,
    theater.no,
    no.planes.parked,
    mog.in.use,
    waiting.mog,
    aircraft.no,
    start.location,
    present.location,
    capacity,
    status,
    type,
    in.use,
    total.on.board,
    and no.missions.flown as integer variables
define route.name,
    location.name,
    and mission as text variables
define route.flight.time,
    total.ac.flight.hours,
    int.total.ac.flight.hours,
    avg.hrs.flown,
    mean.patient.interarrival.time,
        min.batch.size,
        max.batch.size,
        ac.flight.time,
        and int.ac.flight.time as real variables
    define patient.type.mix as an integer, stream 9 variable

temporary entities include patient,
                    travel.leg,
                    aircraft.servicing.a.route,
                    conus.region,
                    org.bed.type,
                    location.3e.feeding.a.4e,
                    mission.delayed

  every patient has a mark.time.3e,
                a mark.time.4e,
                a mark.time.plane,
                a mark.time.conus.asf,
                a patient.type,
                a regulation.status,
                a stabilized.at.this.time,
                a sae.mission,
                a destination,
                a heal.time,
                a hospital.type,
                and may belong to a patient.list
                and may belong to a manifest.list

  every travel.leg has a leg.no,
                a leg.orig,
                a leg.dest,
                a leg.mean.time,
                a dest.reason,
                and belongs to a route.leg.sequence

  every aircraft.servicing.a.route has a ac.servicing.no
                and belongs to a route.aircraft.pool

  every conus.region has a region.number,
                 a region.descriptor,
                 a region.fill.status,
                 a region.theater,
                 and belongs to a region.priority.list
```

```
every org.bed.type has an org.type.number,
                       an org.type.descriptor,
                       an org.theater,
                       and belongs to an org.priority.list

every location.3e.feeding.a.4e has a location.3e.no,
                       and belongs to a location.feeder.pool

every mission.delayed has a mission.make.up,
                          a route.make.up,
                          a pick.up.make.up,
                          a delivery.region.make.up,
                          a theater.make.up
                          and belongs to a mission.delayed.pool

define patient.type,
       regulation.status,
       sae.mission,
       hospital.type,
       destination,
       leg.no,
       leg.orig,
       leg.dest,
       dest.reason,
       ac.servicing.no,
       region.number,
       region.fill.status,
       region.theater,
       org.type.number,
       org.theater,
       location.3e.no,
       mission.make.up,
       route.make.up,
       pick.up.make.up,
       delivery.region.make.up,
       and theater.make.up as integer variables
define region.descriptor,
       and org.type.descriptor as text variables
```

```
define mark.time.3e,
        mark.time.4e,
        mark.time.plane,
        mark.time.conus.asf,
        stabilized.at.this.time,
        heal.time,
        and leg.mean.time as real variables

the system has a stop.time,
           and owns a region.priority.list,
                 an org.priority.list,
                 and a mission.delayed.pool

  define stop.time,
        begin.regulate.time,
        regulate.frequency,
        mean.reconstitute.ac,
        sd.reconstitute.ac,
        min.strat.admin,
        max.strat.admin,
        mean.load.ac,
        mean.unload.ac,
        mean.fuel.ac,
        mean.fly.between.conus.bases,
        cell.fill.policy,
        region.fill.policy,
        begin.heal.time,
        heal.time.frequency,
        theater.evac.policy,
        tot.avg.planes.parked,
        tot.avg.3e.patients,
        and time.incr.int as real variables
 define missions.delayed.because.no.aircraft,
        no.patient.types,
        no.org.bed.types,
        no.conus.regions,
        no.4e.locations,
        no.3e.locations,
        mission.cnt,
        mission.capacity,
        no.ac.types,
        no.theaters,
        time.incr,
        tot.patient.cnt,
    healed.patient.cnt,
    location.patient.type.cnt,
    max.time.incr,
    n.runs,
    runs.counter,
    and clean.up.mission.criteria as integer variables
define strategic.conus.fill.policy,
    aircraft.print.echo,
    route.print.echo,
    location.print.echo,
    regulate.print.echo,
    end.of.run.full.print,
    end.of.run.short.print,
    grand.run.print,
    and bed.print.echo as text variables

define update.mean.arrivals as a 2-dimensional real array
define mean.stabilize.time as a 1-dimensional real array
define std.dev.stabilize.time as a 1-dimensional real array
define mean.heal.time as a 1-dimensional real array
define std.dev.heal.time as a 1-dimensional real array
define patient.type.descriptor as a 1-dimensional text array
define total.beds.available as a 3-dimensional integer array
define total.beds.proj.occupied as a 3-dimensional integer array
define total.beds.occupied as a 3-dimensional integer array
define region.cnt as a 1-dimensional integer array
define region.mission.cnt as a 1-dimensional integer array
define region.mission.flag as a 1-dimensional integer array
define ac.type.flight.hrs as a 1-dimensional real array
define int.ac.type.flight.hrs as a 1-dimensional real array
define ute.rate as a 1-dimensional real array
define int.ute.rate as a 1-dimensional real array
define max.ute.rate as a 1-dimensional real array
define begin.theater.regulate as a 1-dimensional real array
define theater.regulate.frequency as a 1-dimensional real array
define region.fill.capacity as a 1-dimensional integer array
define region.beds.occupied as a 1-dimensional integer array
define heal.count as a 2-dimensional integer array
define check.low demand.cnt as a 1-dimensional real array
define check.low.demand.int as a 1-dimensional real array

define region.priority.list as a fifo set
define org.priority.list as a fifo set
define mission.delayed.pool as a fifo set
```

```
    define not.regulated to mean 0
    define regulated to mean 1
    define regulated.and.mission to mean 2
    define not.full to mean 0
    define full to mean 1
    define schedule.none to mean 0
    define schedule.missions to mean 1
    define idle to mean 0
    define busy to mean 1
    define not.identified to mean 0
    define identified to mean 1
    define load.patients to mean 2
    define unload.patients to mean 3
    define fuel.aircraft.from.conus to mean 4
    define fuel.aircraft.to.conus to mean 5
    define mission.complete to mean 9
    define org.trap to mean no.org.bed.types
    define region.trap to mean no.conus.regions
    define clean.up.mission.from.theater to mean 999
    define no to mean 0
    define yes to mean 1

tally no.routes.flown as the number,
    avg.hours.flown as the mean,
    and total.hours.flown as the sum of route.flight.time
tally total.ac.flight.hours as the sum of ac.flight.time
tally int.total.ac.flight.hours as the sum of int.ac.flight.time
tally no.patients as the number,
    and avg.time.sys.patient as the average of time.in.system
tally no.patients.1 as the number,
    and avg.time.sys.patient.1 as the average of time.in.system.1
tally no.patients.2 as the number,
    and avg.time.sys.patient.2 as the average of time.in.system.2
tally grand.mean.tis as the mean,
    and grand.std.tis as the std.dev of run.tis
tally grand.mean.tis.1 as the mean,
    and grand.std.tis.1 as the std.dev of run.tis.1
tally grand.mean.tis.2 as the mean,
    and grand.std.tis.2 as the std.dev of run.tis.2
tally grand.mean.avg.ute as the mean,
    and grand.std.avg.ute as the std.dev of run.avg.ute
tally grand.mean.max.ute as the mean,
    and grand.std.max.ute as the std.dev of run.max.ute
```

```
tally grand.mean.avg.4e as the mean,
    and grand.std.avg.4e as the std.dev of run.avg.3e
tally grand.mean.avg.planes.parked as the mean,
    and grand.std.avg.planes.parked as the std.dev
    of run.avg.planes.parked
tally grand.mean.pct.patients.transported as the mean,
    and grand.std.pct.patients.transported as the std.dev
    of run.pct.patients.transported
tally grand.mean.pct.missions.delayed as the mean,
    and grand.std.pct.missions.delayed as the std.dev
    of run.pct.missions.delayed
accumulate avg.patients.in.location as the mean,
        and max.patients.in.location as the maximum
        of n.patient.list
accumulate avg.mog.in.use as the mean,
        and max.mog.in.use as the maximum
        of mog.in.use
accumulate avg.waiting.mog as the mean,
        and max.waiting.mog as the maximum
        of waiting.mog
accumulate avg.planes.parked as the mean,
        and max.planes.parked as the maximum
        of no.planes.parked
define time.in.system,
    time.in.system.1,
    time.in.system.2,
    run.tis,
    run.tis.1,
    run.tis.2,
    run.avg.ute,
    run.max.ute,
    run.avg.3e,
    run.avg.planes.parked,
    run.pct.patients.transported,
    and run.pct.missions.delayed as real variables
define hours to mean units
end
```

```
MAIN

"  open 4 for input, name is "pc.dat"
"  use unit 4 for input
"  open 7 for output, name is "pc.out"
"  use unit 7 for output

 call READ.DATA

for runs.counter = 1 to n.runs
 do
  call INITIALIZE
  activate a STOP.SIMULATION in stop.time hours
  start simulation
 loop

stop
end
```

```
EVENT CHECK.DEMAND.FOR.STRAT.AE given THEATER.ID

define I,
    counter,
    pick.up.location.id,
    delivery.region.id,
    mission.id,
    theater.id,
    total.to.assign.to.mission,
    and clean.up.mission.cnt as integer variables

reserve region.cnt as no.conus.regions
reserve region.mission.cnt as no.conus.regions
reserve region.mission.flag as no.conus.regions
reserve check.low.demand.cnt as no.theaters
reserve check.low.demand.int as no.theaters

add 1 to check.low.demand.cnt(theater.id)

for each location with mission(location) = "facility.3e" and
             theater.no(location) = theater.id
do

 for each location.3e.feeding.a.4e in location.feeder.pool(location)
  do
   for every patient in
    patient.list(location.3e.no(location.3e.feeding.a.4e)) with
      regulation.status(patient) = regulated
     do
      add 1 to region.cnt(destination(patient))
     loop
  loop

for counter = 1 to no.conus.regions
 do
  let region.mission.cnt(counter) =
       trunc.f(region.cnt(counter) / mission.capacity)
  if region.mission.cnt(counter) ge 1
   let region.mission.flag(counter) = schedule.missions
  else
   let region.mission.flag(counter) = schedule.none
  always
loop
```

```
      for counter = 1 to no.conus.regions
       do
        if region.mission.flag(counter) = schedule.missions
         for I = 1 to region.mission.cnt(counter)
          do
           let mission.cnt = mission.cnt + 1
           let total.to.assign.to.mission = mission.capacity
            for each location.3e.feeding.a.4e in location.feeder.pool(location)
            until total.to.assign.to.mission = 0
             do
              for each patient in patient.list(location.3e.no(location.3e.feeding.a.4e)),
              with regulation.status(patient) = regulated and destination(patient) = counter,
              until total.to.assign.to.mission = 0
               do
                let regulation.status(patient) = regulated.and.mission
                let sae.mission(patient) = mission.cnt
                let total.to.assign.to.mission = total.to.assign.to.mission-1
               loop
             loop
          let pick.up.location.id = location
          let delivery.region.id = counter
          let mission.id = mission.cnt
          schedule a MISSION.GENERATOR giving pick.up.location.id,
                                              delivery.region.id,
                                               mission.id,
                                              and theater.id now

        loop
      always
     loop
        for counter = 1 to no.conus.regions
          do
            let region.cnt(counter) = 0
            let region.mission.cnt(counter) = 0
            let region.mission.flag(counter) = 0
          loop
 loop


if mod.f(check.low.demand.cnt(theater.id),
               check.low.demand.int(theater.id)) = 0.0

let clean.up.mission.cnt = 0
```

```
for each location with mission(location) = "facility.3e"
                     and theater.no(location) = theater.id,
               until clean.up.mission.cnt = mission.capacity
do

 for each location.3e.feeding.a.4e in location.feeder.pool(location),
  until clean.up.mission.cnt = mission.capacity
   do
    for every patient in patient.list(location.3e.no(location.3e.feeding.a.4e)) with
     regulation.status(patient) = regulated and
     (time.v - mark.time.4e(patient) ge theater.evac.policy),
     until clean.up.mission.cnt = mission.capacity
      do
       add 1 to clean.up.mission.cnt
      loop
  loop
loop

if clean.up.mission.cnt ge clean.up.mission.criteria
 let mission.cnt = mission.cnt + 1
 let clean.up.mission.cnt = 0
 for each location with mission(location) = "facility.3e"
                     and theater.no(location) = theater.id,
               until clean.up.mission.cnt = mission.capacity
do

for each location.3e.feeding.a.4e in location.feeder.pool(location),
 until clean.up.mission.cnt = mission.capacity
  do
   for every patient in patient.list(location.3e.no(location.3e.feeding.a.4e)) with
    regulation.status(patient) = regulated and
    (time.v - mark.time.4e(patient) ge theater.evac.policy),
    until clean.up.mission.cnt = mission.capacity
     do
      add 1 to clean.up.mission.cnt
      let regulation.status(patient) = regulated.and.mission
      let sae.mission(patient) = mission.cnt
     loop
  loop
loop
```

```
   let pick.up.location.id = theater.id
   let delivery.region.id = clean.up.mission.from.theater
   let mission.id = mission.cnt

   schedule a MISSION.GENERATOR giving pick.up.location.id,
                                       delivery.region.id,
                                        mission.id,
                                       and theater.id now

  always

 always
 end
```

```
EVENT CHECK.MISSIONS.DELAYED given AIRCRAFT.ID

define pick.up.location.id,
       route.id,
       aircraft.id,
       mission.id,
       delivery.region.id,
       theater.id,
       and make.up.mission as integer variables
define mission.delayed
       and aircraft.servicing.a.route as pointer variables

let make.up.mission = not.identified

if mission.delayed.pool is not empty
  for every mission.delayed in the mission.delayed.pool,
  until make.up.mission = identified
   do
    for every aircraft.servicing.a.route in
    route.aircraft.pool(route.make.up(mission.delayed)) with
    in.use(ac.servicing.no(aircraft.servicing.a.route)) = yes,
    until make.up.mission = identified
     do
      if aircraft.id = ac.servicing.no(aircraft.servicing.a.route)
       let pick.up.location.id = pick.up.make.up(mission.delayed)
       let route.id = route.make.up(mission.delayed)
       let mission.id = mission.make.up(mission.delayed)
       let delivery.region.id =
           delivery.region.make.up(mission.delayed)
       let theater.id = theater.make.up(mission.delayed)
       let make.up.mission = identified
       remove this mission.delayed from the mission.delayed.pool
       destroy the mission.delayed
      always
     loop
   loop
always
```

```
if make.up.mission = not.identified
 let status(aircraft.id) = idle
else
 activate a FLY.MISSION giving pick.up.location.id, route.id,
                  aircraft.id, mission.id,
                  delivery.region.id and theater.id now
always
end
```

```
ROUTINE to END.OF.RUN

define counter,
      and I, J, and K as integer variables

for every fly.mission in ev.s(i.fly.mission)
  do
   remove this fly.mission from ev.s(i.fly.mission)
   destroy this fly.mission
  loop
for every move.patients.to.4e in ev.s(i.move.patients.to.4e)
  do
   remove this move.patients.to.4e from ev.s(i.move.patients.to.4e)
   destroy this move.patients.to.4e
  loop
for every stop.simulation in ev.s(i.stop.simulation)
  do
   remove this stop.simulation from ev.s(i.stop.simulation)
   destroy this stop.simulation
  loop
for every make.patient in ev.s(i.make.patient)
  do
   remove this make.patient from ev.s(i.make.patient)
   destroy this make.patient
  loop
for every regulate in ev.s(i.regulate)
  do
   remove this regulate from ev.s(i.regulate)
   destroy this regulate
  loop
for every check.demand.for.strat.ae
        in ev.s(i.check.demand.for.strat.ae)
  do
   remove this check.demand.for.strat.ae
       from ev.s(i.check.demand.for.strat.ae)
   destroy this check.demand.for.strat.ae
  loop
for every mission.generator in ev.s(i.mission.generator)
  do
   remove this mission.generator from ev.s(i.mission.generator)
   destroy this mission.generator
  loop
```

```
for every check.missions.delayed in ev.s(i.check.missions.delayed)
 do
  remove this check.missions.delayed
       from ev.s(i.check.missions.delayed)
  destroy this check.missions.delayed
 loop
for every update.parameters in ev.s(i.update.parameters)
 do
  remove this update.parameters from ev.s(i.update.parameters)
  destroy this update.parameters
 loop
for every heal in ev.s(i.heal)
 do
  remove this heal from ev.s(i.heal)
  destroy this heal
 loop

for every location
 do
  for every patient in patient.list(location)
   do
    remove this patient from patient.list(location)
    destroy this patient
   loop
 loop
for every aircraft
 do
  for every patient in manifest.list(aircraft)
   do
    remove this patient from manifest.list(aircraft)
    destroy this patient
   loop
 loop

for every mission.delayed in mission.delayed.pool
 do
  remove this mission.delayed from mission.delayed.pool
  destroy this mission.delayed
 loop
for every route
 do
  let route.flight.time(route) = 0.0
 loop
```

```
for every aircraft
  do
   let status(aircraft) = idle
   let total.on.board(aircraft) = 0
   let ac.flight.time(aircraft) = 0.0
   let int.ac.flight.time(aircraft) = 0.0
   let no.missions.flown(aircraft) = 0
   let total.ac.flight.hours(aircraft) = 0.0
   let int.total.ac.flight.hours(aircraft) = 0.0
   let present.location(aircraft) = start.location(aircraft)
  loop

for every location
 do
  destroy every mog(location.no(location))
  destroy every mog.return(location.no(location))
 loop

let n.mog = n.location
create every mog
for every location
  do
   let u.mog(location) = no.mog(location)
  loop

let n.mog.return = n.location
create every mog.return
for every location
  do
   let u.mog.return(location) = no.mog.return(location)
  loop

for every location
  do
   let no.planes.parked(location) = 0
   let mog.in.use(location) = 0
   let waiting.mog(location) = 0
   let u.mog(location) = no.mog(location)
   let u.mog.return(location) = no.mog.return(location)
  loop
 for every conus.region in the region.priority.list
  do
   let region.fill.status(conus.region) = not.full
  loop

 for I = 1 to no.patient.types
  do
   for J = 1 to no.org.bed.types
    do
     for K = 1 to no.conus.regions
      do
       let total.beds.proj.occupied(I,J,K) = 0
       let total.beds.occupied(I,J,K) = 0
      loop
    loop
  loop

 for counter = 1 to no.conus.regions
  do
   let region.cnt(counter) = 0
   let region.mission.cnt(counter) = 0
   let region.mission.flag(counter) = 0
   let region.beds.occupied(counter) = 0
  loop

 for counter = 1 to no.ac.types
  do
   let ac.type.flight.hrs(counter) = 0.0
   let int.ac.type.flight.hrs(counter) = 0.0
   let ute.rate(counter) = 0.0
   let int.ute.rate(counter) = 0.0
   let max.ute.rate(counter) = 0.0
  loop

 for I = 1 to no.patient.types
  do
   for J = 1 to n.location
    do
     let heal.count(I,J) = 0
    loop
  loop

end
```

```
PROCESS FLY.MISSION given PICK.UP.LOCATION.ID, ROUTE.ID,
                          AIRCRAFT.ID,MISSION.ID,
                          DELIVERY.REGION.ID, and THEATER.ID

define pick.up.location.id,
     route.id,
     aircraft.id,
     mission.id,
     delivery.region.id,
     and theater.id as integer variables
define travel.leg,
     and patient as pointer variables
define flight.time,
     and start.time as a real variables
reserve mean.heal.time as no.patient.types
reserve total.beds.occupied as no.patient.types by no.org.bed.types by
                  no.conus.regions
let flight.time = 0.0

wait uniform.f(min.strat.admin,max.strat.admin,4) hours

if present.location(aircraft.id) ne base.conus.location(route.id)
  let start.time = time.v
  work normal.f(mean.fly.between.conus.bases,
          mean.fly.between.conus.bases*.05,4) hours
  add time.v - start.time to flight.time
  let present.location(aircraft.id) = base.conus.location(route.id)
always

for each travel.leg in route.leg.sequence(route.id)
 do
  let present.location(aircraft.id) = location.no(leg.orig(travel.leg))

  if dest.reason(travel.leg) = fuel.aircraft.from.conus
    add 1 to waiting.mog(leg.dest(travel.leg))
    request 1 unit of mog(leg.dest(travel.leg))
    subtract 1 from waiting.mog(leg.dest(travel.leg))
    add 1 to mog.in.use(leg.dest(travel.leg))
  always
 if dest.reason(travel.leg) = load.patients
   add 1 to waiting.mog(leg.dest(travel.leg))
   request 1 unit of mog(leg.dest(travel.leg))
   relinquish 1 unit of mog(leg.orig(travel.leg))
      activate a MOVE.PATIENTS.TO.4E giving
              mission.id and pick.up.location.id,
              delivery.region.id and theater.id now
   subtract 1 from no.planes.parked(leg.orig(travel.leg))
   subtract 1 from mog.in.use(leg.orig(travel.leg))
   subtract 1 from waiting.mog(leg.dest(travel.leg))
   add 1 to mog.in.use(leg.dest(travel.leg))
 always
 if dest.reason(travel.leg) = fuel.aircraft.to.conus
   request 1 unit of mog.return(leg.dest(travel.leg))
   relinquish 1 unit of mog(leg.orig(travel.leg))
   subtract 1 from no.planes.parked(leg.orig(travel.leg))
   subtract 1 from mog.in.use(leg.orig(travel.leg))
 always

 let start.time = time.v
 work normal.f(leg.mean.time(travel.leg),
          leg.mean.time(travel.leg)*0.05,4) hours
 add time.v - start.time to flight.time

 let present.location(aircraft.id) = location.no(leg.dest(travel.leg))

   if dest.reason(travel.leg) = load.patients
     add 1 to no.planes.parked(leg.dest(travel.leg))
     work normal.f(mean.load.ac,mean.load.ac*.05,4) hours
    for every patient in patient.list(leg.dest(travel.leg)) with
     sae.mission(patient) = mission.id
      do
        remove the patient from patient.list(leg.dest(travel.leg))
        let mark.time.plane(patient) = time.v
        file patient last in manifest.list(aircraft.id)
        add 1 to total.on.board(aircraft.id)
     loop

  always
```

```
if dest.reason(travel.leg) = unload.patients
      work normal.f(mean.unload.ac,mean.unload.ac*.05,4) hours
     for every patient in manifest.list(aircraft.id)
       with destination(patient) = leg.dest(travel.leg)
        do
         remove the patient from manifest.list(aircraft.id)
         let heal.time(patient) = time.v +
            normal.f(mean.heal.time(patient.type(patient)),
                     std.dev.heal.time(patient.type(patient)),4)
         file patient last in patient.list(leg.dest(travel.leg))
         subtract 1 from total.on.board(aircraft.id)
         add 1 to total.beds.occupied(patient.type(patient),
                              hospital.type(patient),
                              leg.dest(travel.leg))
         let time.in.system = time.v - stabilized.at.this.time(patient)
         if theater.id = 1
          let time.in.system.1 =
            time.v - stabilized.at.this.time(patient)
         always
         if theater.id = 2
          let time.in.system.2 =
            time.v - stabilized.at.this.time(patient)
         always
      loop
   always

   if dest.reason(travel.leg) = fuel.aircraft.from.conus
      add 1 to no.planes.parked(leg.dest(travel.leg))
      work normal.f(mean.fuel.ac,mean.fuel.ac*.05,4) hours
   always

   if dest.reason(travel.leg) = fuel.aircraft.to.conus
      work normal.f(mean.fuel.ac,mean.fuel.ac*.05,4) hours
      relinquish 1 unit of mog.return(leg.dest(travel.leg))
   always
```

```
       if dest.reason(travel.leg) = mission.complete
          let present.location(aircraft.id) =
                     location.no(leg.dest(travel.leg))
         until manifest.list(aircraft.id) is empty
          do
           remove first patient from manifest.list(aircraft.id)
           let heal.time(patient) = time.v +
                normal.f(mean.heal.time(patient.type(patient)),
                         std.dev.heal.time(patient.type(patient)),4)
           file the patient last in patient.list(leg.dest(travel.leg))
           subtract 1 from total.on.board(aircraft.id)
             add 1 to total.beds.occupied(patient.type(patient),
                                         hospital.type(patient),
                                         leg.dest(travel.leg))
          let time.in.system = time.v - stabilized.at.this.time(patient)
          if theater.id = 1
            let time.in.system.1 =
              time.v - stabilized.at.this.time(patient)
          always
          if theater.id = 2
            let time.in.system.2 =
              time.v - stabilized.at.this.time(patient)
          always
         loop
      always
  loop

     wait normal.f(mean.reconstitute.ac,
                   sd.reconstitute.ac,4) hours
     let route.flight.time(route.id) = flight.time
     let ac.flight.time(aircraft.id) = flight.time
     let int.ac.flight.time(aircraft.id) = flight.time
     add 1 to no.missions.flown(aircraft.id)

schedule a CHECK.MISSIONS.DELAYED giving AIRCRAFT.ID now

end
```

```
  EVENT HEAL
  define theater id as an integer variable

  reserve heal.count as no.patient.types by n.location
  reserve total.beds.proj.occupied as no.patient.types by
                          no.org.bed.types by
                          no.conus.regions
  reserve total.beds.occupied as no.patient.types by
                          no.org.bed.types by
                          no.conus.regions
  reserve region.beds.occupied as no.conus.regions

  schedule a HEAL given theater.id in heal.time.frequency hours

 for every conus.region in the region.priority.list
     with region.descriptor(conus.region) ne "dummy"
  do
  for every patient in patient.list(region.number(conus.region))
   do
    if time.v ge heal.time(patient)
     remove the patient from patient.list(region.number(conus.region))
     add 1 to healed.patient.cnt
     add 1 to heal.count(patient.type(patient),
                 region.number(conus.region))
     subtract 1 from total.beds.proj.occupied(patient.type(patient),
                          hospital.type(patient),
                          region.number(conus.region))
     subtract 1 from total.beds.occupied(patient.type(patient),
                          hospital.type(patient),
                          region.number(conus.region))
    subtract 1 from region.beds.occupied(region.number(conus.region))
    if region.beds.occupied(region.number(conus.region)) lt
               region.fill.capacity(region.number(conus.region))
     let region.fill.status(conus.region) = not.full
    always
    destroy the patient
   always
  loop
 loop

end
```

```
ROUTINE to INITIALIZE

define location.id,
     counter,
     I,  J,
     and theater.id as an integer variable

let time.v = 0.0

let time.incr = 1

for every location with mission(location) = "facility.3e"
   do
     let location.id = location.no(location)
     schedule a MAKE.PATIENT giving location.id and time.incr
      in exponential.f(mean.patient.interarrival.time(location),1)
      hours

   loop

for counter = 1 to no.theaters
 do
  let theater.id = counter
  schedule a REGULATE giving theater.id in
                begin.theater.regulate(theater.id) hours
  schedule a HEAL giving theater.id in
                begin.heal.time hours
 loop

let mission.cnt = 0
let tot.patient.cnt = 0
let healed.patient.cnt = 0
let tot.avg.planes.parked = 0
let tot.avg.3e.patients = 0
let missions.delayed.because.no.aircraft = 0

reserve region.fill.capacity as no.conus.regions
reserve region.beds.occupied as no.conus.regions
for counter = 1 to no.conus.regions
 do
  let region.fill.capacity(counter) = 0
  let region.beds.occupied(counter) = 0
 loop
for counter = 1 to no.conus.regions
```

```
 do
  for I = 1 to no.patient.types
   do
    for J = 1 to no.org.bed.types
     do
     let region.fill.capacity(counter) = region.fill.capacity(counter)
     + total.beds.available(I,J,counter)
     loop
   loop
 loop

for every route
 do
  reset totals of route.flight.time(route)
 loop

for every aircraft
 do
  reset totals of ac.flight.time(aircraft)
  reset totals of int.ac.flight.time(aircraft)
 loop

reset totals of time.in.system
reset totals of time.in.system.1
reset totals of time.in.system.2

for every location
 do
  reset totals of n.patient.list(location)
  reset totals of mog.in.use(location)
  reset totals of waiting.mog(location)
  reset totals of no.planes.parked(location)
 loop

schedule a UPDATE.PARAMETERS in time.incr.int hours

end
```

```
  EVENT MAKE.PATIENT GIVEN LOCATION.ID and TIME.ID

  define location.id,
       time.id,
       counter,
       and batch.size as an integer variables
  define batch.real,
       and time.to.stabilize as real variables

  reserve mean.stabilize.time as no.patient.types
  reserve std.dev.stabilize.time as no.patient.types

  if time.id = time.incr
  schedule a MAKE.PATIENT giving location.id and time.incr in
   exponential.f(mean.batch.interarrival.time(location.id),1) hours

  let batch.real = uniform.f(min.batch.size(location.id),
                   max.batch.size(location.id),2)
  let batch.size = trunc.f(batch.real)
  for counter = 1 to batch.size
  do
   create a patient
   add 1 to tot.patient.cnt
   let mark.time.3e(patient) = time.v
   let patient.type(patient) = patient.type.mix(location.id)
   let time.to.stabilize =
    normal.f(mean.stabilize.time(patient.type(patient)),
          std.dev.stabilize.time(patient.typc(patient)),3)
   let stabilized.at.this.time(patient) = time.v + time.to.stabilize
   let regulation.status(patient) = not.regulated
   file the patient in patient.list(location.id)
  loop
else
always

end
```

```
EVENT MISSION.GENERATOR given PICK.UP.LOCATION.ID,
                              DELIVERY.REGION.ID,
                              MISSION.ID, and THEATER.ID

 define pick.up.location.id,
     delivery.region.id,
     mission.id,
     theater.id,
     route.id,
     route.resource,
     aircraft.id,
     and min.flight.ptr as integer variables
 define min.flight.time as a real variable
 define aircraft.available.to.fly as a text variable

 let route.resource = not.identified
 let aircraft.available.to.fly = "no"

 if delivery.region.id = clean.up.mission.from.theater
  for each route with route.theater.no(route) = theater.id,
   until route.resource = identified
    do
     if region.destination(route) = delivery.region.id
      let route.resource = identified
      let route.id = route
     always
    loop
else
 for each route, until route.resource = identified
  do
   for each travel.leg in route.leg.sequence(route),
    until route.resource = identified
     do
      if dest.reason(travel.leg) = load.patients and
        leg.dest(travel.leg) = pick.up.location.id and
         region.destination(route) = delivery.region.id
           let route.resource = identified
           let route.id = route
      always
     loop
  loop
always
```

```
if route.resource = not.identified
  print 2 lines with pick.up.location.id and delivery.region.id thus
  ///error in EVENT MISSION.GENERATOR////Route not found
  ///need route w/ pickup at location ** and delivery to region **
always

for each aircraft.servicing.a.route in route.aircraft.pool(route.id),
 with status(ac.servicing.no(aircraft.servicing.a.route)) = idle and
      in.use(ac.servicing.no(aircraft.servicing.a.route)) = yes and
      present.location(ac.servicing.no(aircraft.servicing.a.route)) =
      base.conus.location(route.id),
find the first case
 if found
    let min.flight.time = 10000.00
    for each aircraft.servicing.a.route
    in route.aircraft.pool(route.id), with
    status(ac.servicing.no(aircraft.servicing.a.route)) = idle and
    in.use(ac.servicing.no(aircraft.servicing.a.route)) = yes and
    present.location(ac.servicing.no(aircraft.servicing.a.route)) =
    base.conus.location(route.id),
     do
      if ac.flight.time(ac.servicing.no(aircraft.servicing.a.route))
      le min.flight.time
       let min.flight.time =
            ac.flight.time(ac.servicing.no(aircraft.servicing.a.route))
       let min.flight.ptr = ac.servicing.no(aircraft.servicing.a.route)
      always
    loop
   let aircraft.id = min.flight.ptr
   let status(min.flight.ptr) = busy
   let aircraft.available.to.fly = "yes"
else
```

```
   for each aircraft.servicing.a.route in route.aircraft.pool(route.id),
     with status(ac.servicing.no(aircraft.servicing.a.route)) = idle and
     in.use(ac.servicing.no(aircraft.servicing.a.route)) = yes,
      find the first case
      if found
       let min.flight.time = 10000.00
       for each aircraft.servicing.a.route
       in route.aircraft.pool(route.id), with
       status(ac.servicing.no(aircraft.servicing.a.route)) = idle and
       in.use(ac.servicing.no(aircraft.servicing.a.route)) = yes,
        do
         if ac.flight.time(ac.servicing.no(aircraft.servicing.a.route))
         le min.flight.time
          let min.flight.time =
               ac.flight.time(ac.servicing.no(aircraft.servicing.a.route))
          let min.flight.ptr = ac.servicing.no(aircraft.servicing.a.route)
         always
        loop
      let aircraft.id = min.flight.ptr
      let status(min.flight.ptr) = busy
      let aircraft.available.to.fly = "yes"

     else
        create a mission.delayed
        let pick.up.make.up(mission.delayed) = pick.up.location.id
        let mission.make.up(mission.delayed) = mission.id
        let route.make.up(mission.delayed) = route.id
        let delivery.region.make.up(mission.delayed) =
           delivery.region.id
        let theater.make.up(mission.delayed) = theater.id
        file mission.delayed in mission.delayed.pool
        let aircraft.available.to.fly = "no"
        add 1 to missions.delayed.because.no.aircraft
    always

   always

 if aircraft.available.to.fly = "yes"
   activate a FLY.MISSION giving pick.up.location.id, route.id,
           aircraft.id, mission.id, delivery.region.id,
           and theater.id now
 always

 end
```

```
PROCESS MOVE.PATIENTS.TO.4E given MISSION.ID, PICK.UP.LOCATION.ID,
                                  DELIVERY.REGION.ID, and
                                  THEATER.ID

define mission.id,
     pick.up.location.id,
     delivery.region.id,
     and theater.id as integer variables

define counter as an integer variable
let counter = 0

if delivery.region.id = clean.up.mission.from.theater
  for every location with mission(location) = "facility.3e"
  and theater.no(location) = theater.id
   do
    for every location.3e.feeding.a.4e in
    location.feeder.pool(location)
     do
      for every patient in
      patient.list(location.3e.no(location.3e.feeding.a.4e)) with
      sae.mission(patient) = mission.id
       do
        remove the patient from
        patient.list(location.3e.no(location.3e.feeding.a.4e))
        let mark.time.4e(patient) = time.v
        add 1 to counter
        file the patient last in patient.list(location)
     loop
    loop
  loop
else
```

```
for every location.3e.feeding.a.4e in
location.feeder.pool(pick up location.id)
  do
   for every patient in
   patient.list(location.3e.no(location.3e.feeding.a.4e)) with
   sae.mission(patient) = mission.id
    do
     remove the patient from
     patient.list(location.3e.no(location.3e.feeding.a.4e))
     let mark.time.4e(patient) = time.v
     add 1 to counter
     file the patient last in patient.list(pick.up.location.id)
    loop
  loop

always
end
```

```
ROUTINE to READ.DATA

define counter,
    I, J, and K as integer variables

read aircraft.print.echo,
   route.print.echo,
   location.print.echo,
   regulate.print.echo,
   bed.print.echo,
   end.of.run.full.print,
   end.of.run.short.print,
   grand.run.print

read n.runs,
   stop.time
read time.incr.int,
   max.time.incr

read n.aircraft,
   no.ac.types
 create every aircraft
  for each aircraft,
   do
    read aircraft.no(aircraft),
       start.location(aircraft),
       capacity(aircraft),
       status(aircraft),
       type(aircraft),
       and in.use(aircraft)
    let present.location(aircraft) = start.location(aircraft)
   loop

read mean.reconstitute.ac,
   sd.reconstitute.ac,
   min.strat.admin,
   max.strat.admin,
   mean.load.ac,
   mean.unload.ac,
   mean.fuel.ac,
   mean.fly.between.conus.bases
 read n.route
  create every route
   for each route
    do
     read route.name(route),
        region.destination(route),
        route.theater.no(route),
        base.conus.location(route),
        and no.aircraft.in.route.pool(route)
      for counter = 1 to no.aircraft.in.route.pool(route)
       do
        create an aircraft.servicing.a.route
        read ac.servicing.no(aircraft.servicing.a.route)
        file aircraft.servicing.a route in
           route.aircraft.pool(route)
       loop
     until mode is text,
      do
       create a travel.leg
        read leg.no(travel.leg),
           leg.orig(travel.leg),
           leg.dest(travel.leg),
           leg.mean.time(travel.leg),
           and dest.reason(travel.leg)
       file travel.leg in route.leg.sequence(route)
      loop
   loop
start new card

read no.theaters
read n.location
read no.4e.locations,
   no.3e.locations
reserve update.mean.arrivals as n.location by max.time.incr
 create every location
  for each location
   do
    read location.no(location),
       location.name(location),
       mission(location),
       no.mog(location),
       no.mog.return(location)
     if mission(location) = "facility.3e"
       read theater.no(location),
                    mean.batch.interarrival.time(location),
                    min.batch.size(location),
                    max.batch.size(location)
               for counter = 1 to max.time.incr
                do
                 read update.mean.arrivals(location.no(location),
                                          counter)
                loop
               read patient.type.mix(location)
            always
            if mission(location) = "facility.3e"
             read theater.no(location),
                 no.facility.3e.feeders(location)
              for counter = 1 to no.facility.3e.feeders(location)
               do
                 create a location.3e.feeding.a.4e
                 read location.3e.no(location.3e.feeding.a.4e)
                 file location.3e.feeding.a.4e in
                             location.feeder.pool(location)
               loop
           always
     loop

 let n.mog = n.location
 create every mog
 for every location
   do
    let u.mog(location) = no.mog(location)
   loop
 let n.mog.return = n.location
 create every mog.return
 for every location
   do
    let u.mog.return(location) = no.mog.return(location)
   loop
read no.patient.types
reserve mean.stabilize.time as no.patient.types
reserve std.dev.stabilize.time as no.patient.types
reserve mean.heal.time as no.patient.types
reserve std.dev.heal.time as no.patient.types
reserve patient.type.descriptor as no.patient.types
```

```
for counter = 1 to no.patient.types
   do
     read patient.type.descriptor(counter),
          mean.stabilize.time(counter),
          std.dev.stabilize.time(counter),
          mean.heal.time(counter),
          std.dev.heal.time(counter)
   loop
read begin.heal.time,
    heal.time.frequency

reserve begin.theater.regulate as no.theaters
reserve theater.regulate.frequency as no.theaters
reserve check.low.demand.int as no.theaters
for counter = 1 to no.theaters
 do
  read begin.theater.regulate(counter),
      theater.regulate.frequency(counter),
      check.low.demand.int(counter)
 loop

read cell.fill.policy,
    region.fill.policy,
    strategic.conus.fill.policy,
    mission.capacity,
    theater.evac.policy,
    clean.up.mission.criteria

read no.org.bed.types
for counter = 1 to (no.org.bed.types * no.theaters)
  do
    create an org.bed.type
    read org.type.number(org.bed.type),
        org.type.descriptor(org.bed.type),
        org.theater(org.bed.type)
    file org.bed.type in org.priority.list
  loop

read no.conus.regions
for counter = 1 to (no.conus.regions * no.theaters)
  do
    create a conus.region
    read region.number(conus.region),
        region.descriptor(conus.region),
      region.fill.status(conus.region),
      region.theater(conus.region)
   file conus.region in region.priority.list
  loop

reserve total.beds.available,
     total.beds.occupied,
     and total.beds.proj.occupied
   as no.patient.types by no.org.bed.types by no.conus.regions
for I = 1 to no.patient.types
 do
  for J = 1 to no.org.bed.types
   do
    for K = 1 to no.conus.regions
     do
      read total.beds.available(I,J,K)
     loop
   loop
 loop

start new page

if aircraft.print.echo = "aircraft.echo.on"
 print 5 lines thus
  Echo Input Data for Strategic Aeromedical Evacuation Simulation
  --------------------------------------------------------------

AIRCRAFT STATUS:                status codes -  0-idle
                              1-busy
for each aircraft with in.use(aircraft) = yes
 do
  print 4 lines with aircraft.no(aircraft),
             start.location(aircraft),
             capacity(aircraft),
             status(aircraft),
             and type(aircraft) thus

  Aircraft # ** is originating from location number **
   This aircraft has a capacity of *** patients
   Its current status is **, It is aircraft type *
loop
```

```
 skip 3 lines
  print 11 lines with mean.reconstitute.ac,
                    sd.reconstitute.ac,
                    min.strat.admin,
                    max.strat.admin,
                    mean.load.ac,
                    mean.unload.ac,
                    mean.fuel.ac,
                    and mean.fly.between.conus.bases thus
    Mean time to reconstitute a/c for strategic mission: **.* hrs
    Std dev   "  "    "  "  "    "     "  : **.* hrs

    Min Delay after strat mission requested before takeoff: **.* hrs
    Max  "   "   "    "      "     "     "  : **.* hrs

    Mean time to load patients on aircraft      : **.* hrs
    Mean time to unload patients                : **.* hrs
    Mean time to fuel aircraft at interim stop  : **.* hrs
    Mean time to transfer a/c to other CONUS base: **.* hrs
    (assumes two home bases - one for each theater)
  start new page
  always

if route.print.echo = "route.echo.on"
  print 4 lines thus
    ROUTE DESCRIPTIONS:    Reason for stop codes -  2-load patients
                                                    3-unload patients
                                                    4-fuel aircraft
                                                    9-mission complete
  skip 2 lines
for each route
  do
   print 2 lines with route.name(route),
                    region.destination(route),
                    route.theater.no(route),
                    and base.conus.location(route) thus
   ********************************
   CONUS Region Destination: *  Theater Serviced: *  Home CONUS Base: *
   skip 1 line
   print 2 lines thus
                                              Travel      Reason
   Leg #   Origination #   Destination #   Mean Time   for Stop
   skip 1 line
```

```
for each travel.leg in route.leg.sequence(route)
   print 1 line with leg.no(travel.leg),
                  leg.orig(travel.leg),
                  leg.dest(travel.leg),
                  leg.mean.time(travel.leg),
                  and dest.reason(travel.leg) thus
     *       **            **        **.* hrs        *
    skip 1 line
    print 1 line thus
     The following aircraft are assigned to service this route:
begin report printing
    for each aircraft.servicing.a.route in route.aircraft.pool(route),
       in groups of 15
      print 1 line with a group of
        ac.servicing.no(aircraft.servicing.a.route) fields as follows
        ** ** ** ** ** ** ** ** ** ** ** ** ** ** **
end
    skip 2 lines
  loop
start new page
always

if location.print.echo = "location.echo.on"
  print 6 lines with no.theaters, n.location and no.4e.locations thus
LOCATION INFORMATION:

    The scenario contains * theaters of operation
    There are a total of ** distinct locations among all routes
        ** of these are 4e facilities

print 9 lines thus
    Patient Type Codes -  1-Medicine
                          2-Surgery
                          3-Psychiatric
                          4-Orthopedic
                          5-Burns
                          6-Spinal
                          7-OB/GYN
                          8-Pediatrics

   skip 1 line
```

```
for each location
    do
      print 3 lines with location.no(location),
                  location.name(location),
                  mission(location) thus
   # Name       Mission
   - ---------- ------------
   * ********** ************
skip 1 line
 if mission(location) = "facility.3e"
  print 1 line with theater.no(location) thus
   This facility is located in theater *
  skip 1 line
  print 4 lines with mean.patient.interarrival.time(location),
              min.batch.size(location),
             max.batch.size(location) thus
         Mean Patient   Batch Size
           Int.Time      Min    Max
         ------------   ------- -------
           **.***        ***.*  ***.*
 skip 1 line
 always

 if mission(location) = "facility.3e"
 print 3 lines thus
          Arriving     Cumulative
         Patient Type   Probability

    for each random.e in patient.type.mix(location)
     do
     print 1 line with ivalue.a(random.e) and prob.a(random.e) thus
            **           *.***
    loop
 skip 1 line
 always
```

```
   if mission(location) = "facility.3e"
    print 3 lines thus
            Patients Arrive to location:
            Time Increment   Mean.Patient.Interarrival.Time

      for counter = 1 to max.time.incr
       do
        print 1 line with counter and
          update.mean.arrivals(location.no(location),counter) thus
               **              **** ****
       loop
   always

   if mission(location) = "facility.4e"
    print 5 lines with theater.no(location),
                 no.facility.3e.feeders(location),
                 and u.mog(location.no(location)) thus
     This facility is located in theater *
     This 4th echelon facility receives patients from **
         3rd echelon facilities
     Max on Ground is ** for this facility
     The following 3rd echelon facilities send patients:
  for each location.3e.feeding.a.4e in location.feeder.pool(location)
    print 1 line with location.3e.no(location.3e.feeding.a.4e)
    and location.name(location.3e.no(location.3e.feeding.a.4e)) thus
       ** ***********
  always
     skip 2 lines
    loop
skip 3 lines
print 7 lines thus
STABILIZATION TIMES BY PATIENT TYPE (ALL LOCATIONS):

 Patient  Patient     Mean       Std Dev    Mean      Std Dev
  Type     Type      Stabilize  Stabilize   Heal       Heal
  Code   Descriptor  Tine (Hrs) Time (Hrs) Time(Hrs) Time(Hrs)
-------- ----------- ---------- ---------- --------- ---------
```

```
for counter = 1 to no.patient.types
   do
     print 1 line with counter, patient.type.descriptor(counter),
                    mean.stabilize.time(counter),
                    std.dev.stabilize.time(counter),
                    mean.heal.time(counter),
                    std.dev.heal.time(counter) thus
   *   ************   **.*   **.*   ****.*   ****.*
   loop
skip 2 lines
print 3 lines with begin.heal.time and heal.time.frequency thus
  At sim time *****.* hrs every CONUS patient is checked for discharge
          from Hospital
    This occurs every ***.* hours
start new page
always.

if regulate.print.echo = "regulate.echo.on"
  print 2 lines thus
REGULATE PARAMETERS:

for counter = 1 to no.theaters
  do
   print 3 lines with counter,
                 begin.theater.regulate(counter),
                 theater.regulate.frequency(counter),
                 and check.low.demand.int(counter) thus
    Theater * will begin regulating at sim time *****.* hrs
          and will continue to regulate every ***.* hrs
     A check for low demand will occur every ** regulate cylce
  skip 1 line
 loop
```

```
print 9 lines with cell.fill.policy,
                   region.fill.policy,
                   strategic.conus.fill.policy,
                   no.org.bed.types,
                   no.conus.regions,
                   mission.capacity,
                   theater.evac.policy,
                   and clean.up.mission.criteria thus
   Fill policy is *.** for each cell
   Fill policy is *.** for each region
   Strategic CONUS fill policy is ************************
   There are ** organizational bed types (Mil, VA, NDBS & Dummy)
   There are ** CONUS regions to deliver patients (ASFs)
   Smallest Capacity A/C for Computing # Missions Needed: ****
   Theater Evac Policy: ***.* hours (Cleanup Mission Scheduled)
   Cleanup Mission Criteria: *** patients in the theater exceeding
    theater evac policy
print 2 lines with n.runs and stop.time thus
   Number of replications: **
   Simulation stop time is ****.* hours
skip 2 lines
print 2 lines thus
   The following is the priority order and fill status for regions:
   (fill status=1 means region is full - not available to regulate)
skip 1 line
   for counter = 1 to no.theaters
   do
     print 1 line with counter thus
      Theater *:
     skip 1 line
     for each conus.region in region.priority.list with
           region.theater(conus.region) = counter
     print 1 line with region.number(conus.region),
                  region.descriptor(conus.region),
                  and region.fill.status(conus.region) thus
      Region # **, ***********   Fill Status - *
     skip 2 line
   loop
print 1 line thus
   The following is the priority order for organizational bed type:
skip 1 line
```

```
for counter = 1 to no.theaters
     do
      print 1 line with counter thus
       Theater *:
      skip 1 line
      for each org.bed.type in org.priority.list with
               org.theater(org.bed.type) = counter
      print 1 line with org.type.number(org.bed.type),
                     and org.type.descriptor(org.bed.type) thus
        Org # **, ******
      skip 2 lines
     loop
    skip 3 lines
start new page
always

if bed.print.echo = "bed.echo.on"
  print 8 lines thus
TOTAL BEDS AVAILABLE:

patient type (I)
organization (J)

                              conus  region (K)
               1      2      3      4      5      6      7

for I = 1 to no.patient.types
  do
   for J = 1 to no.org.bed.types
     do
       print 1 line with I,
                         J,
                         total.beds.available(I,J,1),
                         total.beds.available(I,J,2),
                         total.beds.available(I,J,3),
                         total.beds.available(I,J,4),
                         total.beds.available(I,J,5),
                         total.beds.available(I,J,6),
                         and total.beds.available(I,J,7) thus
   I=*, J=* *****  *****  *****  *****  *****  *****  *****
   loop
   skip 1 line
  loop
start new page
```

```
print 8 lines thus
TOTAL BEDS PROJECTED OCCUPIED:

patient type (I)
organization (J)

                         conus  region (K)
          1    2    3    4    5    6    7

for I = 1 to no.patient.types
 do
  for J = 1 to no.org.bed.types
   do
     print 1 line with I,
                       J,
                       total.beds.proj.occupied(I,J,1),
                       total.beds.proj.occupied(I,J,2),
                       total.beds.proj.occupied(I,J,3),
                       total.beds.proj.occupied(I,J,4),
                       total.beds.proj.occupied(I,J,5),
                       total.beds.proj.occupied(I,J,6),
                       and total.beds.proj.occupied(I,J,7) thus
  I=*, J=* *****  *****  *****  *****  *****  *****  *****
  loop
  skip 1 line
 loop
start new page
```

```
  print 8 lines thus
  TOTAL BEDS OCCUPIED:

  patient type (I)
  organization (J)

                    conus  region (K)
          1    2    3    4    5    6    7

 for I = 1 to no.patient.types
  do
   for J = 1 to no.org.bed.types
    do
      print 1 line with I,
                   J,
                   total.beds.occupied(I,J,1),
                   total.beds.occupied(I,J,2),
                   total.beds.occupied(I,J,3),
                   total.beds.occupied(I,J,4),
                   total.beds.occupied(I,J,5),
                   total.beds.occupied(I,J,6),
                   and total.beds.occupied(I,J,7) thus
   I=*, J=* *****  *****  *****  *****  *****  *****  *****
   loop
   skip 1 line
  loop
start new page
always

end
```

```
EVENT REGULATE given THEATER.ID

define theater.id as an integer variable
reserve region.beds.occupied as no.conus.regions
reserve region.fill.capacity as no.conus.regions
reserve total.beds.proj.occupied as no.patient.types by
                                        no.org.bed.types by
                                        no.conus.regions


schedule a REGULATE giving theater.id in
                     theater.regulate.frequency(theater.id) hours

if strategic.conus.fill.policy = "organization.then.region"

for each location with mission(location) = "facility.3e" and
                        theater.no(location) = theater.id
do
  for each patient in patient.list(location) with
    regulation.status(patient) = not.regulated and
    stabilized.at.this.time(patient) le time.v,
      until regulation.status(patient) = regulated
    do
       for each org.bed.type in the org.priority.list with
           org.theater(org.bed.type) = theater.id,
           until regulation.status(patient) = regulated
         do
           for each conus.region in the region.priority.list with
             region.theater(conus.region) = theater.id and
             region.fill.status(conus.region) = not.full,
               until regulation.status(patient) = regulated
             do

if total.beds.proj.occupied(patient.type(patient),
   org.type.number(org.bed.type),region.number(conus.region)) lt
   cell.fill.policy * total.beds.available(patient.type(patient),
   org.type.number(org.bed.type),region.number(conus.region))
      add 1 to total.beds.proj.occupied(patient.type(patient),
            org.type.number(org.bed.type),region.number(conus.region))
      add 1 to region.beds.occupied(region.number(conus.region))
      if region.beds.occupied(region.number(conus.region)) ge
        region.fill.capacity(region.number(conus.region)) *
        region.fill.policy
          let region.fill.status(conus.region) = full
       always
       let regulation.status(patient) = regulated
       le; destination(patient) = region.number(conus.region)
       let hospital.type(patient) = org.type.number(org.bed.type)

   else
            if org.type.number(org.bed.type) = org.trap and
              region.number(conus.region) = region.trap
             print 3 lines thus

          /////////ERROR - make # dummy org beds larger/////////

            always
   always
         loop
        loop
    loop
 loop
always


if strategic.conus.fill.policy = "region.then.organization"


for each location with mission(location) = "facility.3e" and
                  theater.no(location) = theater.id
 do
  for each patient in patient.list(location) with
    regulation.status(patient) = not.regulated and
    stabilized.at.this.time(patient) le time.v,
     until regulation.status(patient) = regulated
   do
      for each conus.region in the region.priority.list with
         region.theater(conus.region) = theater.id and
         region.fill.status(conus.region) = not.full,
         until regulation.status(patient) = regulated
       do
         for each org.bed.type in the org.priority.list with
            org.theater(org.bed.type) = theater.id,
            until regulation.status(patient) = regulated
          do
      if total.beds.proj.occupied(patient.type(patient),
         org.type.number(org.bed.type),region.number(conus.region)) lt
         cell.fill.policy * total.beds.available(patient.type(patient),
         org.type.number(org.bed.type),region.number(conus.region))
           add 1 to total.beds.proj.occupied(patient.type(patient),
                org.type.number(org.bed.type),region.number(conus.region))
           add 1 to region.beds.occupied(region.number(conus.region))
           if region.beds.occupied(region.number(conus.region)) ge
              region.fill.capacity(region.number(conus.region)) *
              region.fill.policy
                let region.fill.status(conus.region) = full
           always
           let regulation.status(patient) = regulated
           let destination(patient) = region.number(conus.region)
           let hospital.type(patient) = org.type.number(org.bed.type)
      else
              if region.number(conus.region) = region.trap and
                 org.type.number(org.bed.type) = org.trap
                 print 3 lines thus

              /////////ERROR - make # dummy region beds larger/////////

              always
      always
            loop
           loop
      loop
    loop
   always

   schedule a CHECK.DEMAND.FOR.STRAT.AE giving theater.id now

   end
```

```
PROCESS STOP.SIMULATION

define counter,
    I,
    J,
    and aircraft.cnt as integer variables

reserve ac.type.flight.hrs as no.ac.types
reserve ute.rate as no.ac.types
reserve total.beds.occupied as no.patient.types by
                    no.org.bed.types by
                    no.conus.regions


if end.of.run.short.print ="end.of.run.short.print.on"
start new page
print 1 line with time.v/24.0 and runs.counter thus
Results after ***.* days of simulation, Replication # **
skip 3 lines

for every location with mission(location) = "facility.3e"
 do
  add avg.patients.in.location(location) to tot.avg.3e.patients
 loop
for every location with mission(location) = "facility.4e"
 do
  add avg.planes.parked(location) to tot.avg.planes.parked
 loop

print 1 line thus
General Information:
skip 1 line
print 16 lines with tot.patient.cnt,
              no.patients,
              no.patients.1,
              no.patients.2,
              avg.time.sys.patient,
              avg.time.sys.patient.1,
              avg.time.sys.patient.2,
              tot.avg.3e.patients/no.3e.locations,
              tot.avg.planes.parked/no.4e.locations,
              missions.delayed.because.no.aircraft thus
```

```
        Total Casualties: ******

        Total Patients Transported from Theater to CONUS: ******
             Theater 1: ******
             Theater 2: ******

        Average Time Patient was in System: ****.* hours
             Avg Time Theater 1: ****.* hours
             Avg Time Theater 2: ****.* hours
        (Stabilized at 3E Facility to Arrival at CONUS Region)

        Avg # Patients in 2E Facilities: ****.

        Avg # Planes Parked at 3E Facilities: **.**

        Total Missions Delayed: ***
    skip 1 line
    always

    if end.of.run.full.print ="end.of.run.full.print.on"
    start new page
    print 1 line thus
    Route Information:
    skip 2 lines
    for each route with no.routes.flown gt 0
    print 4 lines with route.name(route),
               no.routes.flown(route),
               total.hours.flown(route),
               and avg.hours.flown(route) thus
     Route                          Times   Total        Avg Flight Hrs
                                    Flown  Flight Hrs     Per Mission
    *************************       ****    *****.*          **.*

   skip 2 lines

   start new page
   print 2 lines thus
   Disposition of all Patients:
    (some patients may be on a/c)
   skip 1 line
```

```
  for every location with mission(location) = "conus.asf"
    do
      print 5 lines with location,
                    location.name(location),
                    n.patient.list(location),
                    avg.patients.in.location(location),
                    and max.patients.in.location(location) thus
 Location # **, ***********, currently has **** patients
        Avg # in Region: *****.      Max # in Region: *****.

    Patient Type    Current Number
    ------------     --------------
      for counter = 1 to no.patient.types
       do
        let location.patient.type.cnt = 0
        for J = 1 to no.org.bed.types
         do
          let location.patient.type.cnt = location.patient.type.cnt
            + total.beds.occupied(counter,J,location.no(location))
         loop
      print 1 line with patient.type.descriptor(counter)
                   and location.patient.type.cnt thus
    ***********        *****
      loop
       skip 2 lines
   loop
 for every location with mission(location) eq "facility.4e" or
                   mission(location) eq "enroute.fuel"
 do
    if avg.waiting.mog(location) = 0.0
      let max.waiting.mog(location) = 0.0
    always
    print 7 lines with location,
                   location.name(location),
                   n.patient.list(location),
                   avg.patients.in.location(location),
                   max.patients.in.location(location),
                   no.mog(location),
                   u.mog(location),
                   avg.waiting.mog(location),
                   max.waiting.mog(location),
                   avg.mog.in.use(location),
                   max.mog.in.use(location),
                   avg.planes.parked(location),
                and max.planes.parked(location) thus
 Location # **, *************, currently has ***** patients
       Avg # in facility: *****.       Max # in facility: *****.
       Amount of MOG at Location: **
       Amount of MOG Currently Available: **
       Avg Waiting for MOG: **.**     Max Waiting for MOG: **.**
       Avg MOG in use     : **.**     Max MOG in use   : **.**
       Avg Planes Parked  : **.**     Max Planes Parked: **.**
    skip 1 line
 loop
 for every location with mission(location) = "facility.3e"
   do
    print 2 lines with location,
                location.name(location),
                n.patient.list(location),
                avg.patients.in.location(location),
                and max.patients.in.location(location) thus
 Location # **, ************, currently has **** patients
       Avg # in Hospital: *****.     Max # in Hospital: *****.
 skip 1 line
   loop
 start new page
  print 7 lines with healed.patient.cnt thus
 CONUS BEDS STATUS:
 (a total of ****** have recovered and been discharged)
 patient type (I)
 organization (J)

                conus  region (K)
          1    2    3    4    5    6    7
 skip 1 line
 for I = 1 to no.patient.types
  do
   for J = 1 to no.org.bed.types
    do
     print 1 line with I,
                J,
                total.beds.available(I,J,1),
                total.beds.available(I,J,2),
                total.beds.available(I,J,3),
                total.beds.available(I,J,4),
                total.beds.available(I,J,5),
                total.beds.available(I,J,6),
                and total.beds.available(I,J,7) thus
    I=*, J=* *****  *****  *****  *****  *****  *****  *****
    loop
    skip 1 line
   loop
  start new page
  print 8 lines thus
  TOTAL BEDS PROJECTED OCCUPIED:

  patient type (I)
  organization (J)

                   conus  region (K)
          1    2    3    4    5    6    7

 for I = 1 to no.patient.types
   do
    for J = 1 to no.org.bed.types
     do
       print 1 line with I,
                      J,
                      total.beds.proj.occupied(I,J,1),
                      total.beds.proj.occupied(I,J,2),
                      total.beds.proj.occupied(I,J,3),
                      total.beds.proj.occupied(I,J,4),
                      total.beds.proj.occupied(I,J,5),
                      total.beds.proj.occupied(I,J,6),
                      and total.beds.proj.occupied(I,J,7) thus
   I=*, J=* *****  *****  *****  *****  *****  *****  *****
   loop
   skip 1 line
  loop
 start new page
 print 8 lines thus
 TOTAL BEDS OCCUPIED:

 patient type (I)
 organization (J)

                  conus  region (K)
         1    2    3    4    5    6    7
 for I = 1 to no.patient.types
   do
    for J = 1 to no.org.bed.types
      do
         print 1 line with I,
                         J,
                         total.beds.occupied(I,J,1),
                         total.beds.occupied(I,J,2),
                         total.beds.occupied(I,J,3),
                         total.beds.occupied(I,J,4),
                         total.beds.occupied(J,J,5),
                         total.beds.occupied(I,J,6),
                         and total.beds.occupied(I,J,7) thus
    I=*, J=* *****  *****  *****  *****  *****  *****  *****
    loop
    skip 1 line
  loop

 start new page
 print 1 line thus
 AIRCRAFT STATUS:
 skip 2 lines
 for every aircraft
   do
    print 1 line with aircraft,
                 type(aircraft),
                 n.manifest.list(aircraft),
                 status(aircraft),
                 present.location(aircraft),
                 no.missions.flown(aircraft),
                 and total.ac.flight.hours(aircraft) thus
#**, type *, w/ *** on brd, status *, at loc# *, *** msns, ****.* tot hrs
  loop
skip 3 lines
always
```

```
 for counter = 1 to no.ac.types
  do
   let aircraft.cnt = 0
     for every aircraft with type(aircraft) = counter and
                         in.use(aircraft) = yes
      do
       add total.ac.flight.hours(aircraft) to ac.type.flight.hrs(counter)
       add 1 to aircraft.cnt
      loop
   if aircraft.cnt gt 0
    let ute.rate(counter) = ac.type.flight.hrs(counter) /
                      (time.v / 24.0) / real.f(aircraft.cnt)
   else
    let ute.rate(counter) = 0.0
   always

 print 2 lines with aircraft.cnt,
                   counter,
                   ute.rate(counter),
                   time.incr.int,
                   and max.ute.rate(counter) thus
The ** aircraft of type * had an avg utilization rate of **.* hrs per day
     The max ute rate over a ****.* hr period was: **.* hrs per day
skip 1 line
  loop

 let run.tis = avg.time.sys.patient
 let run.tis.1 = avg.time.sys.patient.1
 let run.tis.2 = avg.time.sys.patient.2
 let run.avg.ute = ute.rate(1)
 let run.max.ute = max.ute.rate(1)
 let run.avg.3e = tot.avg.3e.patients / no.3e.locations
let run.avg.planes.parked = tot.avg.planes.parked /
                    no.4e.locations
let run.pct.patients.transported = no.patients /
                         tot.patient.cnt
let run.pct.missions.delayed =
 missions.delayed.because.no.aircraft / mission.cnt
 if runs.counter = n.runs and grand.run.print = "grand.run.print.on"
 print 17 lines with n.runs,
                    grand.mean.tis,
                    grand.std.tis,
                    grand.mean.tis.1,
                    grand.std.tis.1,
                    grand.mean.tis.2,
                    grand.std.tis.2,
                    grand.mean.avg.ute,
                    grand.std.avg.ute,
                    grand.mean.max.ute,
                    grand.std.max.ute,
                    grand.mean.avg.4e,
                    grand.std.avg.4e,
                    grand.mean.avg.planes.parked,
                    grand.std.avg.planes.parked,
                    grand.mean.pct.patients.transported,
                    grand.std.pct.patients.transported,
                    grand.mean.pct.missions.delayed,
                    grand.std.pct.missions.delayed thus
  Final Grand Stats for Simulation Run (** replications)

                              Std.Dev
    Avg Time in System: ****.* hrs          ****.****
     Avg TIS Theater1: ****.* hrs           ****.****
     Avg TIS Theater2: ****.* hrs           ****.****
   Avg Ute Rate on A/C: **.* hrs per day    ****.****
     Max Avg Ute Rate: **.* hrs per day     ****.****
   (10 day period)
   Avg # Patients in
     Field Hospitals: ****.                 ****.****
   Avg Planes Parked
          at APOES: *.***                   ****.****
   Avg % Patients
        Transported: *.***                  ****.****
   Avg % Missions
            Delayed: *.***                  ****.****
always

Call END.OF.RUN

end
```

```
EVENT UPDATE.PARAMETERS

define aircraft.cnt,
     and counter as integer variables

reserve update.mean.arrivals as n.location by max.time.incr
reserve int.ute.rate as no.ac.types
reserve int.ac.type.flight.hrs as no.ac.types
reserve max.ute.rate as no.ac.types


if time.incr lt max.time.incr

schedule an UPDATE.PARAMETERS in time.incr.int hours


add 1 to time.incr
for every location with mission(location) = "facility.3e"
 do
  let mean.patient.interarrival.time(location) =
     update.mean.arrivals(location.no(location),time.incr)
  schedule a MAKE.PATIENT giving location.no(location) and
   time.incr in exponential.f(mean.patient.interarrival.time(location),1)
   hours
 loop

always

if time.incr = 2
print 1 line with runs.counter thus
INTERIM RESULTS for replication # *:
skip 3 lines
always
```

```
 for counter = 1 to no.ac.types
  do
   let aircraft.cnt = 0
   let int.ac.type.flight.hrs(counter) = 0.0
   let int.ute.rate(counter) = 0.0
     for every aircraft with type(aircraft) = counter and
                          in.use(aircraft) = yes
      do
       add int.total.ac.flight.hours(aircraft) to
           int.ac.type.flight.hrs(counter)
       add 1 to aircraft.cnt
      loop
   if aircraft.cnt gt 0
    let int.ute.rate(counter) = int.ac.type.flight.hrs(counter) /
                 (time.incr.int / 24.0) / real.f(aircraft.cnt)
   else
    let int.ute.rate(counter) = 0.0
   always

   if int.ute.rate(counter) ge max.ute.rate(counter)
    let max.ute.rate(counter) = int.ute.rate(counter)
   always

print 2 lines with time.incr - 1,
             time.v,
             aircraft.cnt,
             counter,
             and int.ute.rate(counter) thus
During time increment # *, ending at time = *****.* hrs
The ** aircraft of type * had an avg ute rate of **.* hrs per day
skip 1 line
 loop

for every aircraft
 do
  let int.total.ac.flight.hours(aircraft) = 0.0
 loop

end
```

Appendix B. Scenario Data File

```
aircraft.echo.on
route.echo.on
location.echo.on
regulate.echo.on
bed.echo.on
end.of.run.full.print.on
end.of.run.short.print.on
grand.run.print.on

5  4320.01
240.0  18

45  1
1   2 102 0 1  1
2   2 102 0 1  1
3   2 102 0 1  1
4   2 102 0 1  1
5   2 102 0 1  1
6   2 102 0 1  1
7   2 102 0 1  1
8   2 102 0 1  1
9   2 102 0 1  1
10 2 102 0 1  1
11 2 102 0 1  1
12 2 102 0 1  1
13 2 102 0 1  1
14 2 102 0 1  1
15 2 102 0 1  1
16 2 102 0 1  1
17 2 102 0 1  1
18 6 102 0 1  1
19 6 102 0 1  1
20 6 102 0 1  1
21 6 102 0 1  1
22 6 102 0 1  1
23 6 102 0 1  1
24 6 102 0 1  1
25 6 102 0 1  1
26 6 102 0 1  1
27 6 102 0 1  1
28 6 102 0 1  1
29 6 102 0 1  1
30 6 102 0 1  1
31 6 102 0 1  1
32 6 102 0 1  1
```

```
33 6 102 0 1  1
34 6 102 0 1  1
35 6 102 0 1  1
36 6 102 0 1  1
37 6 102 0 1  1
38 6 102 0 1  1
39 6 102 0 1  1
40 6 102 0 1  1
41 6 102 0 1  1
42 6 102 0 1  1
43 6 102 0 1  1
44 6 102 0 1  1
45 6 102 0 1  1

4.0 0.5
1.0 5.0

3.5
1.5

1.0

5.0

37
Route_1_FromCONUS_ASF_2      1   1  2
45  1  2   3   4   5   6   7  8  9  10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   2 13    7.3    4
            2  13   8   7.9    2
            3    8 13   7.9    5
            4  13   1   7.7    3
            5    1   2   2.0    9
Route_2_FromCONUS_ASF_2      2   1  2
45  1   2  3   4   5   6   7   8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   2 13    7.3    4
            2  13   8   7.9    2
            3    8 13   7.9    5
            4  13   2   7.3    9
```

```
Route_3_FromCONUS_ASF_2       3   1   2
45  1  2  3   4   5   6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1    2 13    7.3    4
            2  13    8    7.9    2
            3    8 13    7.9    5
            4  13    3    9.3    3
            5    3    2    2.0    9
Route_4_FromCONUS_ASF_2       4   1   2
45  1  2   3   4   5  6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1    2 13    7.3    4
            2  13    8    7.9    2
            3    8 13    7.9    5
            4  13    4  10.8    3
            5    4    2    3.0    9
Route_5_FromCONUS_ASF_2       5   1   2
45  1  2   3   4  5  6   7   8   9  10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1    2 13    7.3    4
            2  13    8    7.9    2
            3    8 13    7.9    5
            4  13    5    8.6    3
            5    5    2    3.0    9
Route_6_FromCONUS_ASF_2       6   1   2
45  1  2  3   4   5   6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1    2 13    7.3    4
            2  13    8    7.9    2
            3    8 13    7.9    5
            4  13    6  13.0    3
            5    6    2    5.0    9
Route_7_FromCONUS_ASF_2       7   1   2
45  1  2   3   4   5  6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1    2 13    7.3    4
            2  13    8    7.9    2
            3    8 13    7.9    5
            4  13    7    9.5    3
            5    7    2    3.0    9
```

```
Route_8_FromCONUS_ASF_2      1   1  2
45  1  2  3   4   5  6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
             1    2 13    7.3    4
             2  13   9    8.3    2
             3    9 13    8.3    5
             4  13   1    7.7    3
             5    1   2   2.0    9
Route_9_FromCONUS_ASF_2      2   1  2
45  1  2  3   4   5  6   7  8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
             1    2 13    7.3    4
             2  13   9    8.3    2
             3    9 13    8.3    5
             4  13   2    7.3    9
Route_10_FromCONUS_ASF_2     3   1  2
45  1  2  3   4   5  6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
             1    2 13    7.3    4
             2  13   9    8.3    2
             3    9 13    8.3    5
             4  13   3    9.3    3
             5    3   2   2.0    9
Route_11_FromCONUS_ASF_2     4   1  2
45  1  2  3   4   5  6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
             1    2 13    7.3    4
             2  13   9    8.3    2
             3    9 13    8.3    5
             4  13   4   10.8    3
             5    4   2   3.0    9
Route_12_FromCONUS_ASF_2     5   1  2
45  1  2  3   4   5  6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
             1    2 13    7.3    4
             2  13   9    8.3    2
             3    9 13    8.3    5
             4  13   5    8.6    3
             5    5   2   3.0    9
```

```
Route_13_FromCONUS_ASF_2      6   1   2
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
          1   2 13   7.3   4
          2  13  9   8.3   2
          3   9 13   8.3   5
          4  13  6  13.0   3
          5   6  2   5.0   9
Route_14_FromCONUS_ASF_2      7   1   2
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
          1   2 13   7.3   4
          2  13  9   8.3   2
          3   9 13   8.3   5
          4  13  7   9.5   3
          5   7  2   3.0   9
Route_15_FromCONUS_ASF_2      1   1   2
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
          1   2 13   7.3   4
          2  13 10   6.0   2
          3  10 13   6.0   5
          4  13  1   7.7   3
          5   1  2   2.0   9
Route_16_FromCONUS_ASF_2      2   1   2
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
          1   2 13   7.3   4
          2  13 10   6.0   2
          3  10 13   6.0   5
          4  13  2   7.3   9
Route_17_FromCONUS_ASF_2      3   1   2
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
          1   2 13   7.3   4
          2  13 10   6.0   2
          3  10 13   6.0   5
          4  13  3   9.3   3
          5   3  2   2.0   9
```

```
Route_18_FromCONUS_ASF_2     4   1   2
45  1  2   3   4   5   6  7   8   9 10 11 12 13 14 15 16 17 18 19 20
   21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
   41 42 43 44 45
               1    2 13    7.3    4
               2  13 10    6.0    2
               3  10 13    6.0    5
               4  13   4  10.8    3
               5    4   2    3.0    9
Route_19_FromCONUS_ASF_2     5   1   2
45  1  2   3   4   5   6   7  8   9 10 11 12 13 14 15 16 17 18 19 20
   21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
   41 42 43 44 45
               1    2 13    7.3    4
               2  13 10    6.0    2
               3  10 13    6.0    5
               4  13   5    8.6    3
               5    5   2    3.0    9
Route_20_FromCONUS_ASF_2     6   1   2
45  1  2   3   4   5   6   7  8   9 10 11 12 13 14 15 16 17 18 19 20
   21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
   41 42 43 44 45
               1    2 13    7.3    4
               2  13 10    6.0    2
               3  10 13    6.0    5
               4  13   6  13.0    3
               5    6   2    5.0    9
Route_21_FromCONUS_ASF_2     7   1   2
45  1  2   3   4   5  6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
   21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
   41 42 43 44 45
               1    2 13    7.3    4
               2  13 10    6.0    2
               3  10 13    6.0    5
               4  13   7    9.5    3
               5    7   2    3.0    9
Route_22_FromCONUS_ASF_6     1   2   6
45  1  2   3   4  5  6   7   8   9  10 11 12 13 14 15 16 17 18 19 20
   21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
   41 42 43 44 45
               1    6 14    4.0    4
               2  14 11    6.9    2
               3  11 14    6.9    5
               4  14   1    8.0    3
               5    1   6    5.0    9
```

```
Route_23_FromCONUS_ASF_6     2  2  6
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14   4.0   4
            2  14 11   6.9   2
            3  11 14   6.9   5
            4  14  2   8.5   3
            5   2  6   5.0   9
Route_24_FromCONUS_ASF_6     3  2  6
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14   4.0   4
            2  14 11   6.9   2
            3  11 14   6.9   5
            4  14  3   9.0   3
            5   3  6   5.0   9
Route_25_FromCONUS_ASF_6     4  2  6
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14   4.0   4
            2  14 11   6.9   2
            3  11 14   6.9   5
            4  14  4   6.1   3
            5   4  6   2.0   9
Route_26_FromCONUS_ASF_6     5  2  6
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14   4.0   4
            2  14 11   6.9   2
            3  11 14   6.9   5
            4  14  5   5.8   3
            5   5  6   3.0   9
Route_27_FromCONUS_ASF_6     6  2  6
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14   4.0   4
            2  14 11   6.9   2
            3  11 14   6.9   5
            4  14  6   4.0   9
```

```
Route_28_FromCONUS_ASF_6      7   2   6
45  1  2   3   4   5   6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14    4.0    4
            2  14 11    6.9    2
            3  11 14    6.9    5
            4  14   7    7.4    3
            5    7   6    3.0    9
Route_29_FromCONUS_ASF_6      1   2   6
45  1  2   3   4   5   6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14    4.0    4
            2  14 12    8.1    2
            3  12 14    8.1    5
            4  14   1    8.0    3
            5    1   6    5.0    9
Route_30_FromCONUS_ASF_6      2   2   6
45  1  2   3   4   5   6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14    4.0    4
            2  14 12    8.1    2
            3  12 14    8.1    5
            4  14   2    8.5    3
            5    2   6    5.0    9
Route_31_FromCONUS_ASF_6      3   2   6
45  1  2   3   4   5   6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14    4.0    4
            2  14 12    8.1    2
            3  12 14    8.1    5
            4  14   3    9.0    3
            5    3   6    5.0    9
Route_32_FromCONUS_ASF_6      4   2   6
45  1  2   3   4   5   6   7   8   9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14    4.0    4
            2  14 12    8.1    2
            3  12 14    8.1    5
            4  14   4    6.1    3
            5    4   6    2.0    9
```

```
Route_33_FromCONUS_ASF_6      5   2   6
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14   4.0   4
            2  14 12   8.1   2
            3  12 14   8.1   5
            4  14  5   5.8   3
            5   5  6   3.0   9
Route_34_FromCONUS_ASF_6      6   2   6
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14   4.0   4
            2  14 12   8.1   2
            3  12 14   8.1   5
            4  14  6   4.0   9
Route_35_FromCONUS_ASF_6      7   2   6
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   6 14   4.0   4
            2  14 12   8.1   2
            3  12 14   8.1   5
            4  14  7   7.4   3
            5   7  6   3.0   9
Route_36_FromCONUS_ASF_2      999  1   2
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
            1   2 13   7.3   4
            2  13  8   7.9   2
            3   8  9   1.0   2
            4   9 10   1.0   2
            5  10 13   6.0   5
            6  13  2   7.3   3
            7   2  1   1.0   3
            8   1  3   2.0   3
            9   3  5   2.0   3
           10   5  4   2.0   3
           11   4  6   2.0   3
           12   6  7   1.0   3
           13   7  2   5.0   9
```

```
Route_37_FromCONUS_ASF_6      999  2   6
45  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
  41 42 43 44 45
          1   6 14   4.0   4
          2  14 11   6.9   2
          3  11 12   1.0   2
          4  12 14   8.1   5
          5  14  6   4.0   3
          6   6  4   2.0   3
          7   4  5   2.0   3
          8   5  1   2.0   3
          9   1  2   1.0   3
         10   2  3   1.5   3
         11   3  7   1.0   3
         12   7  6   5.0   9

end

2

24  5 10
1  McGuire          conus.asf    0  0
2  Andrews          conus.asf    0  0
3  Charleston       conus.asf    0  0
4  Kelly            conus.asf    0  0
5  Scott            conus.asf    0  0
6  Norton           conus.asf    0  0
7  DummyRegion      conus.asf    0  0
8  SWA_APOE_1       facility.3e  3   0    1    2  15 16
9  SWA_APOE_2       facility.3e  3   0    1    2  17 18
10 SWA_APOE_3       facility.3e  3   0    1    2  19 20
11 FE_APOE_1        facility.3e  3   0    2    2  21 22
12 FE_APOE_2        facility.3e  3   0    2    2  23 24
13 SWA_INT          enroute.fuel 12 30
14 FE_INT           enroute.fuel 12 30
15 SWA_HOSP_1       facility.2e    0  0   1  2160.0 5.0  25.9999
2160.0000  440.8163   86.7470   30.7692    21.6000   10.8000
    8.6365    9.8226   6.9700     21.6000    21.6000    21.6000
9999.0000 9999.0000 9999.0000 9999.0000 9999.0000 9999.0000
 0.126 1 0.441 2 0.032 3 0.368 4
 0.026 5 0.007 6 0.000 7 0.000 8 *
```

```
16 SWA_HOSP_2   facility.2e  0 0   1  2160.0 5.0  25.9999
 2160.0000   440.8163    86.7470    30.7692     21.6000    10.8000
    8.6365     9.8226     6.9700    21.6000     21.6000    21.6000
 9999.0000 9999.0000 9999.0000 9999.0000 9999.0000 9999.0000
  0.126 1 0.441 2 0.032 3 0.368 4
  0.026 5 0.007 6 0.000 7 0.000 8 *
17 SWA_HOSP_3   facility.2e  0 0   1  2160.0 5.0  25.9999
 2160.0000   440.8163    86.7470    30.7692     21.6000    10.8000
    8.6365     9.8226     6.9700    21.6000     21.6000    21.6000
 9999.0000 9999.0000 9999.0000 9999.0000 9999.0000 9999.0000
  0.126 1 0.441 2 0.032 3 0.368 4
  0.026 5 0.007 6 0.000 7 0.000 8 *
18 SWA_HOSP_4   facility.2e  0 0   1  2160.0 5.0  25.9999
 2160.0000   440.8163    86.7470    30.7692     21.6000    10.8000
    8.6365     9.8226     6.9700    21.6000     21.6000    21.6000
 9999.0000 9999.0000 9999.0000 9999.0000 9999.0000 9999.0000
  0.126 1 0.441 2 0.032 3 0.368 4
  0.026 5 0.007 6 0.000 7 0.000 8 *
19 SWA_HOSP_5   facility.2e  0 0   1  2160.0 5.0  25.9999
 2160.0000   440.8163    86.7470    30.7692    21.6000     10.8000
    8.6365     9.8226     6.9700    21.6000    21.6000     21.6000
 9999.0000 9999.0000 9999.0000 9999.0000 9999.0000 9999.0000
  0.126 1 0.441 2 0.032 3 0.368 4
  0.026 5 0.007 6 0.000 7 0.000 8 *
20 SWA_HOSP_6   facility.2e  0 0   1  2160.0 5.0  25.9999
 2160.0000   440.8163    86.7470    30.7692     21.6000    10.8000
    8.6365     9.8226     6.9700    21.6000     21.6000    21.6000
 9999.0000 9999.0000 9999.0000 9999.0000 9999.0000 9999.0000
  0.126 1 0.441 2 0.032 3 0.368 4
  0.026 5 0.007 6 0.000 7 0.000 8 *
21 FE_HOSP_1    facility.2e  0 0   2  9999.0 5.0  25.9999
 9999.0000 9999.0000 9999.0000 9999.0000  180.0000   37.8947
   24.0000    16.4009     7.2000      6.5395     6.0050     5.5342
    6.5395     8.9888     8.9888      8.988 8  10.5882   10.5882
  0.126 1 0.441 2 0.032 3 0.368 4
  0.026 5 0.007 6 0.000 7 0.000 8 *
22 FE_HOSP_2    facility.2e  0 0   2  9999.0 5.0  25.9999
 9999.0000 9999.0000 9999.0000 9999.0000  180.0000   37.8947
   24.0000    16.4009     7.2000      6.5395     6.0050     5.5342
    6.5395     8.9888     8.9888      8.9888    10.5882   10.5882
  0.126 1 0.441 2 0.032 3 0.368 4
  0.026 5 0.007 6 0.000 7 0.000 8 *
```

```
23 FE_HOSP_3   facility.2e  0 0   2  9999.0 5.0  25.9999
 9999.0000 9999.0000 9999.0000 9999.0000  42.6036   8.9888
    6.0050      4.0932      1.8000      1.6360     1.5000   1.3846
    1.6360      2.2493      2.2493      2.2493     2.6461   2.6461
  0.126 1 0.441 2 0.032 3 0.368 4
  0.026 5 0.007 6 0.000 7 0.000 8 *
24 FE_HOSP_4   facility.2e  0 0   2  9999.0 5.0  25.9999
 9999.0000 9999.0000 9999.0000 9999.0000  42.6036   8.9888
    6.0050      4.0932      1.8000      1.6360     1.5000   1.3846
    1.6360      2.2493      2.2493      2.2493     2.6461   2.6461
  0.126 1 0.441 2 0.032 3 0.368 4
  0.026 5 0.007 6 0.000 7 0.000 8 *

8
Medicine       6.0    1.0     384.0   19.2
Surgery        6.0    1.0     696.0   34.8
Psychiatric    6.0    1.0     576.0   28.8
Orthopedic    12.0    1.0    1200.0   60.0
Burns         12.0    1.0     792.0   39.6
Spinal        24.0    1.0     912.0   45.6
OB/GYN         6.0   1.0      720.0   36.0
Pediatrics     6.0   1.0      720.0   36.0
240.0  24.0

8.0   8.0  12.0
12.0 8.0  12.0

0.90 0.80  organization.then.region  102  168.0   25

4
1 DOD          1
2 VA           1
3 NDMS         1
4 dummy        1
1 DOD          2
2 VA           2
3 NDMS         2
4 dummy        2
```

```
7
1 Northeast          0 1
2 MidAtlantic        0 1
3 Southeast          0 1
5 Midwest            0 1
4 Southwest          0 1
6 West               0 1
7 dummy              0 1
6 West               0 2
4 Southwest          0 2
5 Midwest            0 2
3 Southwest          0 2
2 MidAtlantic        0 2
1 Northeast          0 2
7 dummy              0 2

4825 1830 2465 2500 1765 1445 0
4455 1685 2275 2300 1630 1330 0
9280 3515 4745 4805 3400 2780 0
   0    0    0    0    0    0 100000
3660 1465 2100 1930 1605 1800 0
3375 1355 1935 1775 1485 1660 0
7035 2820 4040 3710 3090 3460 0
   0    0    0    0    0    0 100000
 970  510  835  660  605  570 0
 895  470  775  610  560  525 0
1865  980 1610 1270 1165 1100 0
   0    0    0    0    0    0 100000
 800  420  990  790  520  565 0
 745  385  910  725  480  520 0
1545  805 1905 1520 1005 1085 0
   0    0    0    0    0    0 100000
 115   60  160   20   35  125 0
 105   55  145   15   30  115 0
 220  115  310   40   70  245 0
   0    0    0    0    0    0 100000
 190   65  175   95   75  180 0
 170   55  165   90   70  165 0
 360  125  340  185  150  345 0
   0    0    0    0    0    0 100000
   0    0    0    0    0    0
   0    0    0    0    0    0
   0    0    0    0    0    0
   0    0    0    0    0    0 100000
```

```
0   0   0   0   0   0
0   0   0   0   0   0
0   0   0   0   0   0
0   0   0   0   0   0 100000
```

Appendix C. Echo of Scenario Data File

```
Echo Input Data for Strategic Aeromedical Evacuation Simulation
----------- ----------------------------------------------------------------

AIRCRAFT STATUS:               status codes -  0-idle
                                               1-busy

Aircraft #  1 is originating from location number  2
 This aircraft has a capacity of 102 patients
 Its current status is  0, It is aircraft type 1

Aircraft #  2 is originating from location number  2
 This aircraft has a capacity of 102 patients
 Its current status is  0, It is aircraft type 1

Aircraft #  3 is originating from location number  2
 This aircraft has a capacity of 102 patients
 Its current status is  0, It is aircraft type 1
```

(information on all aircraft is not shown)

```
.

Aircraft # 44 is originating from location number  6
 This aircraft has a capacity of 102 patients
 Its current status is  0, It is aircraft type 1

Aircraft # 45 is originating from location number  6
 This aircraft has a capacity of 102 patients
 Its current status is  0, It is aircraft type 1


Mean time to reconstitute a/c for strategic mission:  4.0 hrs
Std dev   "   "    "          "   "    "      "    :  0.5 hrs

Min Delay after strat mission requested before takeoff:  1.0 hrs
Max   "     "    "        "      "      "        "    :  5.0 hrs

Mean time to load patients on aircraft          :  3.5 hrs
Mean time to unload patients                    :  1.5 hrs
Mean time to fuel aircraft at interim stop      :  1.0 hrs
Mean time to transfer a/c to other CONUS base   :  5.0 hrs
(assumes two home bases - one for each theater)
```

ROUTE DESCRIPTIONS: Reason for stop codes -
2-load patients
3-unload patients
4-fuel aircraft
9-mission complete

Route_1_FromCONUS_ASF_2
CONUS Region Destination: 1 Theater Serviced: 1 Home CONUS Base: 2

			Travel	Reason
Leg #	Origination #	Destination #	Mean Time	for Stop
1	2	13	7.3 hrs	4
2	13	8	7.9 hrs	2
3	8	13	7.9 hrs	5
4	13	1	7.7 hrs	3
5	1	2	2.0 hrs	9

The following aircraft are assigned to service this route:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40 41 42 43 44 45

(information on all routes is not shown)

Route_34_FromCONUS_ASF_6
CONUS Region Destination: 6 Theater Serviced: 2 Home CONUS Base: 6

			Travel	Reason
Leg #	Origination #	Destination #	Mean Time	for Stop
1	6	14	4.0 hrs	4
2	14	12	8.1 hrs	2
3	12	14	8.1 hrs	5
4	14	6	4.0 hrs	9

The following aircraft are assigned to service this route:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40 41 42 43 44 45

```
LOCATION INFORMATION:

The scenario contains 2 theaters of operation
There are a total of 24 distinct locations among all routes
    5 of these are 4e facilities

Patient Type Codes -  1-Medicine
                      2-Surgery
                      3-Psychiatric
                      4-Orthopedic
                      5-Burns
                      6-Spinal
                      7-OB/GYN
                      8-Pediatrics


# Name         Mission
- ----------   ------------
1 McGuire      conus.asf

.
```

. *(information on all locations is not shown)*

```
.


 # Name           Mission
 - ----------     ------------
7 DummyRegion     conus.asf


# Name            Mission
- ----------      ------------
8 SWA_APOE_1      facility.4e

This facility is located in theater 1
This 4th echelon facility receives patients from  2
    3rd echelon facilities
Max on Ground is  3 for this facility
The following 3rd echelon facilities send patients:
   15  SWA_HOSP_1
   16  SWA_HOSP_2
 # Name           Mission
 - ----------     ------------
 9 SWA_APOE_2  facility.4e

 This facility is located in theater 1
 This 4th echelon facility receives patients from  2
    3rd echelon facilities
 Max on Ground is  3 for this facility
 The following 3rd echelon facilities send patients:
   17  SWA_HOSP_3
   18  SWA_HOSP_4
```

.

. *(information on all locations is not shown)*

.

```
 # Name           Mission
 - ----------     ------------
12 FE_APOE_2   facility.4e

This facility is located in theater 2
This 4th echelon facility receives patients from  2
   3rd echelon facilities
Max on Ground is  3 for this facility
The following 3rd echelon facilities send patients:
  23  FE_HOSP_3
  24  FE_HOSP_4
```

```
 # Name           Mission
 - ----------     ------------
13  SWA_INT     enroute.fuel
```

```
 # Name           Mission
 - ----------     ------------
14  FE_INT      enroute.fuel
```

```
# Name          Mission
- -----------   ------------
15 SWA_HOSP_1  facility.3e

This facility is located in theater 1

      Mean Patient   Batch Size
       Int.Time      Min   Max
      ------------   ----  ----
      2160.000       5.0   26.0

       Arriving    Cumulative
      Patient Type  Probability

           1          0.126
           2          0.567
           3          0.599
           4          0.967
           5          0.993
           6          1.000
           7          1.000
           8          1.000

     Patients Arrive to location:
     Time Increment  Mean.Patient.Interarrival.Time

           1            2160.0000
           2             440.8163
           3              86.7470
           4              30.7692
           5              21.6000
           6              10.8000
           7               8.6365
           8               9.8226
           9               6.9700
          10              21.6000
          11              21.6000
          12              21.6000
          13            9999.0000
          14            9999.0000
          15            9999 0000
          16            9999.0000
          17            9999.0000
          18            9999.0000
```

```
# Name          Mission
- -----------   ------------
24 FE_HOSP_4    facility.3e

This facility is located in theater 2

      Mean Patient    Batch Size
       Int.Time       Min   Max
      ------------    ----  ----
      9999.000        5.0   26.0

       Arriving     Cumulative
      Patient Type  Probability

          1           0.126
          2           0.567
          3           0.599
          4           0.967
          5           0.993
          6           1.000
          7           1.000
          8           1.000

     Patients Arrive to location:
     Time Increment   Mean.Patient.Interarrival.Time

          1             9999.0000
          2             9999.0000
          3             9999.0000
          4             9999.0000
          5               42.6036
          6               8 9888
          7               6.0050
          8               4.0932
          9               1.8000
         10               1.6360
         11               1.5000
         12               1.3846
         13               1.6360
         14               2.2493
         15               2.2493
         16               2.2493
         17               2.6461
         18               2.6461
```

STABILIZATION TIMES BY PATIENT TYPE (ALL LOCATIONS):

Patient Type Code	Patient Type Descriptor	Mean Stabilize Time (Hrs)	Std Dev Stabilize Time (Hrs)	Mean Heal Time(Hrs)	Std Dev Heal Time(Hrs)
1	Medicine	6.0	1.0	6.0	384.0
2	Surgery	6.0	1.0	6.0	696.0
3	Psychiatric	6.0	1.0	6.0	576.0
4	Orthopedic	12.0	1.0	12.0	1200.0
5	Burns	12.0	1.0	12.0	792.0
6	Spinal	24.0	1.0	24.0	912.0
7	OB/GYN	12.0	1.0	6.0	720.0
8	Pediatrics	6.0	1.0	6.0	720.0

At sim time 240.0 hrs every CONUS patient is checked for discharge
from Hospital
This occurs every 24.0 hoours

REGULATE PARAMETERS:

Theater 1 will begin regulating at sim time 8.0 hrs
and will continue to regulate every 8.0 hrs
A check for low demand will occur every 12 regulate cycles

Theater 2 will begin regulating at sim time 12.0 hrs
and will continue to regulate every 8.0 hrs
A check for low demand will occur every 12 regulate cycles

Fill policy is 0.90 for each cell
Fill policy is 0.80 for each region
Strategic CONUS fill policy is organization.then.region
There are 4 organizational bed types (Mil, VA, NDPS & Dummy)
There are 7 CONUS regions to deliver patients (ASFs)
Smallest Capacity A/C for Computing # Missions Needed: 102
Theater Evac Policy: 168.0 hours (Cleanup Mission Scheduled)
Cleanup Mission Criteria: 25 patients in the theater exceeding
theater evac policy
Number of replications: 5
Simulation stop time is 4320.0 hours

The following is the priority order and fill status for regions:
(fill status=1 means region is full - not available to regulate)

Theater 1:

Region # 1, Northeast	Fill Status - 0
Region # 2, MidAtlantic	Fill Status - 0
Region # 3, Southeast	Fill Status - 0
Region # 5, Midwest	Fill Status - 0
Region # 4, Southwest	Fill Status - 0
Region # 6, West	Fill Status - 0
Region # 7, dummy	Fill Status - 0

Theater 2:

Region # 6, West	Fill Status - 0
Region # 4, Southwest	Fill Status - 0
Region # 5, Midwest	Fill Status - 0
Region # 3, Southwest	Fill Status - 0
Region # 2, MidAtlantic	Fill Status - 0
Region # 1, Northeast	Fill Status - 0
Region # 7, dummy	Fill Status - 0

The following is the priority order for organizational bed type:

Theater 1:

Org # 1, DOD
Org # 2, VA
Org # 3, NDMS
Org # 4, dummy

Theater 2:

Org # 1, DOD
Org # 2, VA
Org # 3, NDMS
Org # 4, dummy

TOTAL BEDS AVAILABLE:
patient type (I)
organization (J)

	conus region (K)						
	1	2	3	4	5	6	7
I=1, J=1	4825	1830	2465	2500	1765	1445	0
I=1, J=2	4455	1685	2275	2300	1630	1330	0
I=1, J=3	9280	3515	4745	4805	3400	2780	0
I=1, J=4	0	0	0	0	0	0	100000
I=2, J=1	3660	1465	2100	1930	1605	1800	0
I=2, J=2	3375	1355	1935	1775	1485	1660	0
I=2, J=3	7035	2820	4040	3710	3090	3460	0
I=2, J=4	0	0	0	0	0	0	100000
I=3, J=1	970	510	835	660	605	570	0
I=3, J=2	895	470	775	610	560	525	0
I=3, J=3	1865	980	1610	1270	1165	1100	0
I=3, J=4	0	0	0	0	0	0	100000
I=4, J=1	800	420	990	790	520	565	0
I=4, J=2	745	385	910	725	480	520	0
I=4, J=3	1545	805	1905	1520	1005	1085	0
I=4, J=4	0	0	0	0	0	0	100000
I=5, J=1	115	60	160	20	35	125	0
I=5, J=2	105	55	145	15	30	115	0
I=5, J=3	220	115	310	40	70	245	0
I=5, J=4	0	0	0	0	0	0	100000
I=6, J=1	190	65	175	95	75	180	0
I=6, J=2	170	55	165	90	70	165	0
I=6, J=3	360	125	340	185	150	345	0
I=6, J=4	0	0	0	0	0	0	100000
I=7, J=1	0	0	0	0	0	0	0
I=7, J=2	0	0	0	0	0	0	0
I=7, J=3	0	0	0	0	0	0	0
I=7, J=4	0	0	0	0	0	0	100000
I=8, J=1	0	0	0	0	0	0	0
I=8, J=2	0	0	0	0	0	0	0
I=8, J=3	0	0	0	0	0	0	0
I=8, J=4	0	0	0	0	0	0	100000

TOTAL BEDS PROJECTED OCCUPIED:
patient type (I)
organization (J)

	conus region (K)						
	1	2	3	4	5	6	7
I=1, J=1	0	0	0	0	0	0	0
I=1, J=2	0	0	0	0	0	0	0
I=1, J=3	0	0	0	0	0	0	0
I=1, J=4	0	0	0	0	0	0	0
I=2, J=1	0	0	0	0	0	0	0
I=2, J=2	0	0	0	0	0	0	0
I=2, J=3	0	0	0	0	0	0	0
I=2, J=4	0	0	0	0	0	0	0
I=3, J=1	0	0	0	0	0	0	0
I=3, J=2	0	0	0	0	0	0	0
I=3, J=3	0	0	0	0	0	0	0
I=3, J=4	0	0	0	0	0	0	0
I=4, J=1	0	0	0	0	0	0	0
I=4, J=2	0	0	0	0	0	0	0
I=4, J=3	0	0	0	0	0	0	0
I=4, J=4	0	0	0	0	0	0	0
I=5, J=1	0	0	0	0	0	0	0
I=5, J=2	0	0	0	0	0	0	0
I=5, J=3	0	0	0	0	0	0	0
I=5, J=4	0	0	0	0	0	0	0
I=6, J=1	0	0	0	0	0	0	0
I=6, J=2	0	0	0	0	0	0	0
I=6, J=3	0	0	0	0	0	0	0
I=6, J=4	0	0	0	0	0	0	0
I=7, J=1	0	0	0	0	0	0	0
I=7, J=2	0	0	0	0	0	0	0
I=7, J=3	0	0	0	0	0	0	0
I=7, J=4	0	0	0	0	0	0	0
I=8, J=1	0	0	0	0	0	0	0
I=8, J=2	0	0	0	0	0	0	0
I=8, J=3	0	0	0	0	0	0	0
I=8, J=4	0	0	0	0	0	0	0

TOTAL BEDS OCCUPIED:
patient type (I)
organization (J)

	conus region (K)						
	1	2	3	4	5	6	7
I=1, J=1	0	0	0	0	0	0	0
I=1, J=2	0	0	0	0	0	0	0
I=1, J=3	0	0	0	0	0	0	0
I=1, J=4	0	0	0	0	0	0	0
I=2, J=1	0	0	0	0	0	0	0
I=2, J=2	0	0	0	0	0	0	0
I=2, J=3	0	0	0	0	0	0	0
I=2, J=4	0	0	0	0	0	0	0
I=3, J=1	0	0	0	0	0	0	0
I=3, J=2	0	0	0	0	0	0	0
I=3, J=3	0	0	0	0	0	0	0
I=3, J=4	0	0	0	0	0	0	0
I=4, J=1	0	0	0	0	0	0	0
I=4, J=2	0	0	0	0	0	0	0
I=4, J=3	0	0	0	0	0	0	0
I=4, J=4	0	0	0	0	0	0	0
I=5, J=1	0	0	0	0	0	0	0
I=5, J=2	0	0	0	0	0	0	0
I=5, J=3	0	0	0	0	0	0	0
I=5, J=4	0	0	0	0	0	0	0
I=6, J=1	0	0	0	0	0	0	0
I=6, J=2	0	0	0	0	0	0	0
I=6, J=3	0	0	0	0	0	0	0
I=6, J=4	0	0	0	0	0	0	0
I=7, J=1	0	0	0	0	0	0	0
I=7, J=2	0	0	0	0	0	0	0
I=7, J=3	0	0	0	0	0	0	0
I=7, J=4	0	0	0	0	0	0	0
I=8, J=1	0	0	0	0	0	0	0
I=8, J=2	0	0	0	0	0	0	0
I=8, J=3	0	0	0	0	0	0	0
I=8, J=4	0	0	0	0	0	0	0

Appendix D. Simulation Output

The following are results from replication #5 of the baseline run:

```
INTERIM RESULTS for replication # 5:

During time increment # 1, ending at time =  240.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  1.4 hrs per day

During time increment # 2, ending at time =  480.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  0.6 hrs per day

During time increment # 3, ending at time =  720.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  0.3 hrs per day

During time increment # 4, ending at time =  960.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  0.5 hrs per day

During time increment # 5, ending at time = 1200.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  0.8 hrs per day

During time increment # 6, ending at time = 1440.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  1.6 hrs per day

During time increment # 7, ending at time = 1680.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  2.4 hrs per day

During time increment # 8, ending at time = 1920.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  2.9 hrs per day

During time increment # 9, ending at time = 2160.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  4.6 hrs per day

During time increment #10, ending at time = 2400.0 hrs
The 45 aircraft of type 1 nad an avg ute rate of  4.5 hrs per day

During time increment #11, ending at time = 2640.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  4.8 hrs per day

During time increment #12, ending at time = 2880.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  4.5 hrs per day

During time increment #13, ending at time = 3120.0 hrs
The 45 aircraft of type 1 had an avg ute rate of  4.1 hrs per day
```

During time increment #14, ending at time = 3360.0 hrs
The 45 aircraft of type 1 had an avg ute rate of 3.2 hrs per day

During time increment #15, ending at time = 3600.0 hrs
The 45 aircraft of type 1 had an avg ute rate of 2.7 hrs per day

During time increment #16, ending at time = 3840.0 hrs
The 45 aircraft of type 1 had an avg ute rate of 2.3 hrs per day

During time increment #17, ending at time = 4080.0 hrs
The 45 aircraft of type 1 had an avg ute rate of 2.2 hrs per day

During time increment #18, ending at time = 4320.0 hrs
The 45 aircraft of type 1 had an avg ute rate of 2.0 hrs per day

Results after 180.0 days of simulation, Replication # 5

General Information:

Total Casualties: 71212

Total Patients Transported from Theater to CONUS: 69781
Theater 1: 14940
Theater 2: 54841

Average Time Patient was in System: 73.0 hours
Avg Time Theater 1: 107.6 hours
Avg Time Theater 2: 63.6 hours
(Stabilized at 3E Facility to Arrival at CONUS Region)

Avg # Patients in 2E Facilities: 88

Avg # Planes Parked at 3E Facilities: 0.13

Total Missions Delayed: 0

Route Inforrnation:

Route	Times Flown	Total Flight Hrs	Avg Flight Hrs Per Mission
Route_1_FromCONUS_ASF_2	27	880.5	32.6
Route_2_FromCONUS_ASF_2	1	29.8	29.8
Route_3_FromCONUS_ASF_2	2	67.7	33.9
Route_8_FromCONUS_ASF_2	33	1101.9	33.4
Route_9_FromCONUS_ASF_2	3	94.2	31.4
Route_10_FromCONUS_ASF_2	4	138.7	34.7
Route_15_FromCONUS_ASF_2	37	1070.5	28.9
Route_16_FromCONUS_ASF_2	4	104.1	26.0
Route_17_FromCONUS_ASF_2	6	181.0	30.2
Route_19_FromCONUS_ASF_2	1	29.3	29.3
Route_22_FromCONUS_ASF_6	1	30.9	30.9
Route_23_FromCONUS_ASF_6	1	29.1	29.1
Route_24_FromCONUS_ASF_6	9	289.0	32.1
Route_25_FromCONUS_ASF_6	13	334.2	25.7
Route_26_FromCONUS_ASF_6	7	186.6	26.7
Route_27_FromCONUS_ASF_6	41	892.6	21.8
Route_29_FromCONUS_ASF_6	24	787.5	32.8
Route_30_FromCONUS_ASF_6	22	746.5	33.9
Route_31_FromCONUS_ASF_6	74	2535.1	34.3
Route_32_FromCONUS_ASF_6	89	2515.5	28.3

Route_33_FromCONUS_ASF_6	55	1591.8	28.9
Route_34_FromCONUS_ASF_6	166	4005.5	24.1
Route_36_FromCONUS_ASF_2	30	1355.4	45.2
Route_37_FromCONUS_ASF_6	36	1393.8	38.7

Disposition of all Patients:
(some patients may be on a/c)

Location # 1, McGuire , currently has 1197 patients
Avg # in Region: 2674. Max # in Region: 5554.

Patient Type	Current Number
Medicine	0
Surgery	0
Psychiatric	0
Orthopedic	1179
Burns	18
Spinal	0
OB/GYN	0
Pediatrics	0

Location # 2, Andrews , currently has 739 patients
Avg # in Region: 785. Max # in Region: 2358.

Patient Type	Current Number
Medicine	0
Surgery	0
Psychiatric	0
Orthopedic	723
Burns	16
Spinal	0
OB/GYN	0
Pediatrics	0

```
Location #  3, Charleston , currently has 2181 patients
      Avg # in Region:  2109.      Max # in Region:  5385.

   Patient Type   Current Number
   ------------    --------------
   Medicine              0
   Surgery             466
   Psychiatric           0
   Orthopedic         1597
   Burns               118
   Spinal                0
   OB/GYN                0
   Pediatrics            0


Location #  4, Kelly      , currently has 2403 patients
      Avg # in Region:  2195.      Max # in Region:  4559.

   Patient Type   Current Number
   ------------    --------------
   Medicine              0
   Surgery            1158
   Psychiatric           0
   Orthopedic         1227
   Burns                18
   Spinal                0
   OB/GYN                0
   Pediatrics            0


Location #  5, Scott      , currently has 2229 patients
      Avg # in Region:  1391.      Max # in Region:  3214.

   Patient Type   Current Number
   ------------    --------------
   Medicine              0
   Surgery            1411
   Psychiatric           0
   Orthopedic          786
   Burns                32
   Spinal                0
   OB/GYN                0
   Pediatrics            0
```

```
Location #  6, Norton     , currently has 4050 patients
      Avg # in Region:  3276.      Max # in Region:  5502.

    Patient Type   Current Number
    ------------   --------------
    Medicine          626
    Surgery          1500
    Psychiatric       267
    Orthopedic       1437
    Burns             115
    Spinal            105
    OB/GYN              0
    Pediatrics          0


Location #  7, DummyRegion, currently has    0 patients
      Avg # in Region:     0.     Max # in Region:     0.

    Patient Type   Current Number
    ------------   ------------------
    Medicine            0
    Surgery             0
    Psychiatric         0
    Orthopedic          0
    Burns               0
    Spinal              0
    OB/GYN              0
    Pediatrics          0

Location #  8, SWA_APOE_1    , currently has     0 patients
      Avg # in facility:    13.     Max # in facility:   204.
      Amount of MOG at Location:  3
      Amount of MOG Currently Available:  3
      Avg Waiting for MOG:  0.        Max Waiting for MOG:  0.
      Avg MOG in use      :  0.16     Max MOG in use    :  2.00
      Avg Planes Parked   :  0.05     Max Planes Parked:  2.00

Location #  9, SWA_APOE_2    , currently has     0 patients
      Avg # in facility:    14.     Max # in facility:   204.
      Amount of MOG at Location:  3
      Amount of MOG Currently Available:  3
      Avg Waiting for MOG:  0.        Max Waiting for MOG:  0.
      Avg MOG in use      :  0.14     Max MOG in use    :  2.00
      Avg Planes Parked   :  0.06     Max Planes Parked:  2.00
```

```
Location # 10, SWA_APOE_3   , currently has    0 patients
        Avg # in facility:   12.      Max # in facility:  204.
        Amount of MOG at Location:  3
        Amount of MOG Currently Available:  3
        Avg Waiting for MOG:  0.       Max Waiting for MOG:  0.
        Avg MOG in use     :  0.14    Max MOG in use   :  2.00
        Avg Planes Parked  :  0.06    Max Planes Parked:  2.00

Location # 11, FE_APOE_1    , currently has    0 patients
        Avg # in facility:   26.      Max # in facility:  306.
        Amount of MOG at Location:  3
        Amount of MOG Currently Available:  3
        Avg Waiting for MOG:  0.       Max Waiting for MOG:  0.
        Avg MOG in use     :  0.27    Max MOG in use   :  3.00
        Avg Planes Parked  :  0.09    Max Planes Parked:  3.00

Location # 12, FE_APOE_2    , currently has  204 patients
        Avg # in facility:  120.      Max # in facility:  326.
        Amount of MOG at Location:  3
        Amount of MOG Currently Available:  1
        Avg Waiting for MOG:  0.06     Max Waiting for MOG:  3.00
        Avg MOG in use     :  1.20    Max MOG in use   :  3.00
        Avg Planes Parked  :  0.38    Max Planes Parked:  3.00

Location # 13, SWA_INT      , currently has    0 patients
        Avg # in facility:    0.      Max # in facility:    0.
        Amount of MOG at Location: 12
        Amount of MOG Currently Available: 12
        Avg Waiting for MOG:  0.       Max Waiting for MOG:  0.
        Avg MOG in use     :  0.28    Max MOG in use   :  5.00
        Avg Planes Parked  :  0.03    Max Planes Parked:  3.00

Location # 14, FE_INT       , currently has    0 patients
        Avg # in facility:    0.      Max # in facility:    0.
        Amount of MOG at Location: 12
        Amount of MOG Currently Available: 11
        Avg Waiting for MOG:  0.       Max Waiting for MOG:  0.
        Avg MOG in use     :  0.68    Max MOG in use   :  4.00
        Avg Planes Parked  :  0.18    Max Planes Parked:  4.00

Location # 15, SWA_HOSP_1   , currently has    0 patients
        Avg # in Hospital:   30.    Max # in Hospital:   181.
```

```
Location # 16, SWA_HOSP_2  , currently has    0 patients
      Avg # in Hospital:   36.    Max # in Hospital:  223.

Location # 17, SWA_HOSP_3  , currently has    0 patients
      Avg # in Hospital:   39.    Max # in Hospital:  265.

Location # 18, SWA_HOSP_4  , currently has   17 patients
      Avg # in Hospital:   61.    Max # in Hospital:  294.

Location # 19, SWA_HOSP_5  , currently has   16 patients
      Avg # in Hospital:   52.    Max # in Hospital:  222.

Location # 20, SWA_HOSP_6  , currently has    0 patients
      Avg # in Hospital:   83.    Max # in Hospital:  362.

Location # 21, FE_HOSP_1   , currently has   95 patients
      Avg # in Hospital:   61.    Max # in Hospital:  261.

Location # 22, FE_HOSP_2   , currently has  122 patients
      Avg # in Hospital:  123.    Max # in Hospital:  429.

Location # 23, FE_HOSP_3   , currently has  253 patients
      Avg # in Hospital:  164.    Max # in Hospital:  490.

Location # 24, FE_HOSP_4   , currently has  418 patients
      Avg # in Hospital:  237.    Max # in Hospital:  647.
```

CONUS BEDS STATUS: (a total of 56982 have recovered and been discharged)
patient type (I)
organization (J)

	conus region (K)						
	1	2	3	4	5	6	7
I=1, J=1	4825	1830	2465	2500	1765	1445	0
I=1, J=2	4455	1685	2275	2300	1630	1330	0
I=1, J=3	9280	3515	4745	4805	3400	2780	0
I=1, J=4	0	0	0	0	0	0	100000
I=2, J=1	3660	1465	2100	1930	1605	1800	0
I=2, J=2	3375	1355	1935	1775	1485	1660	0
I=2, J=3	7035	2820	4040	3710	3090	3460	0
I=2, J=4	0	0	0	0	0	0	100000
I=3, J=1	970	510	835	660	605	570	0
I=3, J=2	895	470	775	610	560	525	0
I=3, J=3	1865	980	1610	1270	1165	1100	0
I=3, J=4	0	0	0	0	0	0	100000
I=4, J=1	800	420	990	790	520	565	0
I=4, J=2	745	385	910	725	480	520	0
I=4, J=3	1545	805	1905	1520	1005	1085	0
I=4, J=4	0	0	0	0	0	0	100000
I=5, J=1	115	60	160	20	35	125	0
I=5, J=2	105	55	145	15	30	115	0
I=5, J=3	220	115	310	40	70	245	0
I=5, J=4	0	0	0	0	0	0	100000
I=6, J=1	190	65	175	95	75	180	0
I=6, J=2	170	55	165	90	70	165	0
I=6, J=3	360	125	340	185	150	345	0
I=6, J=4	0	0	0	0	0	0	100000
I=7, J=1	0	0	0	0	0	0	0
I=7, J=2	0	0	0	0	0	0	0
I=7, J=3	0	0	0	0	0	0	0
I=7, J=4	0	0	0	0	0	0	100000
I=8, J=1	0	0	0	0	0	0	0
I=8, J=2	0	0	0	0	0	0	0
I=8, J=3	0	0	0	0	0	0	0
I=8, J=4	0	0	0	0	0	0	100000

TOTAL BEDS PROJECTED OCCUPIED:
patient type (I)
organization (J)

	conus region (K)						
	1	2	3	4	5	6	7
I=1, J=1	2	0	0	0	0	747	0
I=1, J=2	0	0	0	0	0	0	0
I=1, J=3	0	0	0	0	0	0	0
I=1, J=4	0	0	0	0	0	0	0
I=2, J=1	18	0	498	1505	1411	1573	0
I=2, J=2	0	0	0	0	0	0	0
I=2, J=3	0	0	0	0	0	0	0
I=2, J=4	0	0	0	0	0	0	0
I=3, J=1	1	0	0	0	0	300	0
I=3, J=2	0	0	0	0	0	0	0
I=3, J=3	0	0	0	0	0	0	0
I=3, J=4	0	0	0	0	0	0	0
I=4, J=1	716	374	864	685	447	509	0
I=4, J=2	589	346	818	638	431	430	0
I=4, J=3	5	82	2	0	0	589	0
I=4, J=4	0	0	0	0	0	0	0
I=5, J=1	21	25	142	18	32	112	0
I=5, J=2	0	0	0	0	0	8	0
I=5, J=3	0	0	0	0	0	0	0
I=5, J=4	0	0	0	0	0	0	0
I=6, J=1	0	0	0	0	0	110	0
I=6, J=2	0	0	0	0	0	0	0
I=6, J=3	0	0	0	0	0	0	0
I=6, J=4	0	0	0	0	0	0	0
I=7, J=1	0	0	0	0	0	0	0
I=7, J=2	0	0	0	0	0	0	0
I=7, J=3	0	0	0	0	0	0	0
I=7, J=4	0	0	0	0	0	0	0
I=8, J=1	0	0	0	0	0	0	0
I=8, J=2	0	0	0	0	0	0	0
I=8, J=3	0	0	0	0	0	0	0
I=8, J=4	0	0	0	0	0	0	0

TOTAL BEDS OCCUPIED:
patient type (I)
organization (J)

	conus region (K)						
	1	2	3	4	5	6	7
I=1, J=1	0	0	0	0	0	626	0
I=1, J=2	0	0	0	0	0	0	0
I=1, J=3	0	0	0	0	0	0	0
I=1, J=4	0	0	0	0	0	0	0
I=2, J=1	0	0	466	1158	1411	1500	0
I=2, J=2	0	0	0	0	0	0	0
I=2, J=3	0	0	0	0	0	0	0
I=2, J=4	0	0	0	0	0	0	0
I=3, J=1	0	0	0	0	0	267	0
I=3, J=2	0	0	0	0	0	0	0
I=3, J=3	0	0	0	0	0	0	0
I=3, J=4	0	0	0	0	0	0	0
I=4, J=1	682	297	777	612	355	506	0
I=4, J=2	492	344	818	615	431	342	0
I=4, J=3	5	82	2	0	0	589	0
I=4, J=4	0	0	0	0	0	0	0
I=5, J=1	18	16	118	18	32	107	0
I=5, J=2	0	0	0	0	0	8	0
I=5, J=3	0	0	0	0	0	0	0
I=5, J=4	0	0	0	0	0	0	0
I=6, J=1	0	0	0	0	0	105	0
I=6, J=2	0	0	0	0	0	0	0
I=6, J=3	0	0	0	0	0	0	0
I=6, J=4	0	0	0	0	0	0	0
I=7, J=1	0	0	0	0	0	0	0
I=7, J=2	0	0	0	0	0	0	0
I=7, J=3	0	0	0	0	0	0	0
I=7, J=4	0	0	0	0	0	0	0
I=8, J=1	0	0	0	0	0	0	0
I=8, J=2	0	0	0	0	0	0	0
I=8, J=3	0	0	0	0	0	0	0
I=8, J=4	0	0	0	0	0	0	0

```
AIRCRAFT STATUS:

# 1, type 1, w/   0 on brd, status 1, at loc# 2,   9 msns,  339.3 tot hrs
# 2, type 1, w/   0 on brd, status 0, at loc# 2,   4 msns,  132.6 tot hrs
# 3, type 1, w/   0 on brd, status 0, at loc# 2,  11 msns,  376.0 tot hrs
# 4, type 1, w/   0 on brd, status 0, at loc# 2,  15 msns,  496.7 tot hrs
# 5, type 1, w/   0 on brd, status 0, at loc# 2,  21 msns,  688.0 tot hrs
.
.  (information on all aircraft is not shown)
.
#33, type 1, w/   0 on brd, status 0, at loc# 6,  31 msns,  899.1 tot hrs
#34, type 1, w/   0 on brd, status 0, at loc# 6,  12 msns,  317.2 tot hrs
#35, type 1, w/   0 on brd, status 0, at loc# 6,   3 msns,   83.6 tot hrs
#36, type 1, w/   0 on brd, status 1, at loc# 7,  35 msns, 1000.3 tot hrs
#37, type 1, w/   0 on brd, status 0, at loc# 6,   2 msns,   63.1 tot hrs
#38, type 1, w/   0 on brd, status 0, at loc# 6,   8 msns,  209.3 tot hrs
#39, type 1, w/   0 on brd, status 1, at loc#14,  45 msns, 1233.7 tot hrs
#40, type 1, w/ 102 on brd, status 1, at loc#14,  45 msns, 1249.1 tot hrs
#41, type 1, w/   0 on brd, status 0, at loc# 6,   2 msns,   63.8 tot hrs
#42, type 1, w/   0 on brd, status 1, at loc# 6,  31 msns,  890.8 tot hrs
#43, type 1, w/   0 on brd, status 0, at loc# 6,  12 msns,  335.2 tot hrs
#44, type 1, w/   0 on brd, status 0, at loc# 6,  25 msns,  772.9 tot hrs
#45, type 1, w/   0 on brd, status 0, at loc# 6,   2 msns,   64.9 tot hrs

The 45 aircraft of type 1 had an avg utilization rate of  2.5 hrs per day
    The max ute rate over a  240.0 hr period was:  4.8 hrs per day

 Final Grand Stats for Simulation Run ( 5 replications)
                                                Std.Dev
    Avg Time in System:   73.1 hrs              1.1447
     Avg TIS Theater1:  104.2 hrs               4.9994
     Avg TIS Theater2:   64.5 hrs               0.7424
   Avg Ute Rate on A/C:  2.5 hrs per day        0.0725
     Max Avg Ute Rate:  5.0 hrs per day         0.2414
     (10 day period)
    Avg # Patients in
     Field Hospitals:     86.                   3.6833
    Avg Planes Parked
          at APOES:     0.126                   0.0037
      Avg % Patients
        Transported:    0.982                   0.0012
      Avg % Missions
           Delayed: 0.                          0.
```

Appendix E. Casualty Arrivals & Bed Availability for the Two-Theater Scenario

Patient Genera[illegible]es - Two Theater Scenario, Southwest Asia (3 APOEs) & Far East (2 APOEs), 180 Day War, Each APOE has Two 3E Facilities Sending It Patients

Patient Generation Table - Southwest Asia Portion of Scenario
3 APOEs

Total 3E Facilities: 6
Mean Batch Size: 15

M.B.I.T	Days	Medical	%	Surgery	%	Psych	%	Ortho	%	Burns	%	Spinal	%	Total
2160.0000	0-10	1	10.0	5	50.0	0	0.0	4	40.0	0	0.0	0	0.0	10
440.8163	10-20	6	12.2	22	44.9	1	2.0	19	38.8	1	2.0	0	0.0	49
86.7470	20-30	31	12.4	110	44.2	8	3.2	92	36.9	6	2.4	2	0.8	249
30.7692	30-40	89	12.7	309	44.0	22	3.1	258	36.8	19	2.7	5	0.7	702
21.6000	40-50	126	12.6	441	44.1	32	3.2	368	36.8	26	2.6	7	0.7	1000
10.8000	50-60	252	12.6	882	44.1	64	3.2	736	36.8	52	2.6	14	0.7	2000
8.6365	60-70	315	12.6	1103	44.1	80	3.2	920	36.8	65	2.6	18	0.7	2501
9.8226	70-80	277	12.6	970	44.1	70	3.2	810	36.8	57	2.6	15	0.7	2199
6.9700	80-90	390	12.6	1367	44.1	99	3.2	1141	36.8	80	2.6	22	0.7	3099
21.6000	90-100	126	12.6	441	44.1	32	3.2	368	36.8	26	2.6	7	0.7	1000
21.6000	100-110	126	12.6	441	44.1	32	3.2	368	36.8	26	2.6	7	0.7	1000
21.6000	110-120	126	12.6	441	44.1	32	3.2	368	36.8	26	2.6	7	0.7	1000
	Total	1865	12.6	6532	44.1	472	3.2	5452	36.8	384	2.6	104	0.7	14809

Patient Generation Table - Far East Portion of Scenario - APOE_1

Total 3E Facilities: 2
Mean Batch Size: 15

M.B.I.T.	Days	Medical	%	Surgery	%	Psych	%	Ortho	%	Burns	%	Spinal	%	Total
	0-10	0		0		0		0		0		0		0
	10-20	0		0		0		0		0		0		0
	20-30	0		0		0		0		0		0		0
	30-40	0		0		0		0		0		0		0
180.0000	40-50	5	12.5	18	45.0	1	2.5	15	37.5	1	2.5	0	0.0	40
37.8947	50-60	25	13.2	89	46.8	6	3.2	64	33.7	5	2.6	1	0.5	190
24.0000	60-70	38	12.7	132	44.0	10	3.3	110	36.7	8	2.7	2	0.7	300
16.4009	70-80	55	12.5	194	44.2	14	3.2	162	36.9	11	2.5	3	0.7	439
7.2000	80-90	126	12.6	441	44.1	32	3.2	368	36.8	26	2.6	7	0.7	1000
6.5395	90-100	139	12.6	485	44.1	35	3.2	405	36.8	29	2.6	8	0.7	1101
6.0050	100-110	151	12.6	529	44.1	39	3.3	441	36.8	31	2.6	8	0.7	1199
5.5342	110-120	164	12.6	574	44.1	41	3.2	479	36.8	34	2.6	9	0.7	1301
6.5395	120-130	139	12.6	485	44.1	35	3.2	405	36.8	29	2.6	8	0.7	1101
8.9888	130-140	101	12.6	353	44.1	25	3.1	295	36.8	21	2.6	6	0.7	801
8.9888	140-150	101	12.6	353	44.1	25	3.1	295	36.8	21	2.6	6	0.7	801
8.9888	150-160	101	12.6	353	44.1	25	3.1	295	36.8	21	2.6	6	0.7	801
10.5882	160-170	86	12.6	300	44.1	21	3.1	250	36.8	18	2.6	5	0.7	680
10.5882	170-180	86	12.6	300	44.1	21	3.1	250	36.8	18	2.6	5	0.7	680
	Total	1317	12.6	4606	44.1	330	3.2	3834	36.7	273	2.6	74	0.7	10434

Patient Generation Table - Far East Portion of Scenario - APOE_2 Total 3E Facilities: 2
Mean Batch Size: 15

M.B.I.T.	Days	Medical	%	Surgery	%	Psych	%	Ortho	%	Burns	%	Spinal	%	Total
	0-10	0		0		0		0		0		0		0
	10-20	0		0		0		0		0		0		0
	20-30	0		0		0		0		0		0		0
	30-40	0		0		0		0		0		0		0
42.6036	40-50	20	11.8	80	47.3	5	3.0	59	34.9	4	2.4	1	0.6	169
8.9888	50-60	101	12.6	353	44.1	25	3.1	295	36.8	21	2.6	6	0.7	801
6.0050	60-70	151	12.6	529	44.1	39	3.3	441	36.8	31	2.6	8	0.7	1199
4.0932	70-80	221	12.6	776	44.1	56	3.2	648	36.8	46	2.6	12	0.7	1759
1.8000	80-90	504	12.6	1764	44.1	128	3.2	1472	36.8	104	2.6	28	0.7	4000
1.6360	90-100	555	12.6	1940	44.1	141	3.2	1619	36.8	115	2.6	31	0.7	4401
1.5000	100-110	605	12.6	2116	44.1	154	3.2	1766	36.8	125	2.6	34	0.7	4800
1.3846	110-120	655	12.6	2294	44.1	166	3.2	1914	36.8	135	2.6	36	0.7	5200
1.6360	120-130	555	12.6	1940	44.1	141	3.2	1619	36.8	115	2.6	31	0.7	4401
2.2493	130-140	404	12.6	1411	44.1	102	3.2	1178	36.8	84	2.6	22	0.7	3201
2.2493	140-150	404	12.6	1411	44.1	102	3.2	1178	36.8	84	2.6	22	0.7	3201
2.2493	150-160	404	12.6	1411	44.1	102	3.2	1178	36.8	84	2.6	22	0.7	3201
2.6461	160-170	343	12.6	1200	44.1	87	3.2	1001	36.8	71	2.6	19	0.7	2721
2.6461	170-180	343	12.6	1200	44.1	87	3.2	1001	36.8	71	2.6	19	0.7	2721
	Total	5265	12.6	18425	44.1	1335	3.2	15369	36.8	1090	2.6	291	0.7	41775

Patient Generation Table - Totals for the Two Theater Scenario

8447	12.6	29563	44.1	2137	3.2	24655	36.8	1747	2.6	469	0.7	67018

CONUS Hospital Beds

		CONUS Regions						
		1	2	3	4	5	6	
Medical	DOD	4825	1830	2465	2500	1765	1445	14830
	VA	4455	1685	2275	2300	1630	1330	13675
	NDMS	9280	3515	4745	4805	3400	2780	28525
	Total	18560	7030	9485	9605	6795	5555	57030
Surgery	DOD	3660	1465	2100	1930	1605	1800	12560
	VA	3375	1355	1935	1775	1485	1660	11585
	NDMS	7035	2820	4040	3710	3090	3460	24155
	Total	14070	5640	8075	7415	6180	6920	48300
Psychiatric	DOD	970	510	835	660	605	570	4150
	VA	895	470	775	610	560	525	3835
	NDMS	1865	980	1610	1270	1165	1100	7990
	Total	3730	1960	3220	2540	2330	2195	15975
Orthopedic	DOD	800	420	990	790	520	565	4085
	VA	745	385	910	725	480	520	3765
	NDMS	1545	805	1905	1520	1005	1085	7865
	Total	3090	1610	3805	3035	2005	2170	15715
Burns	DOD	115	60	160	20	35	125	515
	VA	105	55	145	15	30	115	465
	NDMS	220	115	310	40	70	245	1000
	Total	440	230	615	75	135	485	1980
Spinal	DOD	190	65	175	95	75	180	780
	VA	170	55	165	90	70	165	715
	NDMS	360	125	340	185	150	345	1505
	Total	720	245	680	370	295	690	3000
Totals	DOD	10560	4350	6725	5995	4605	4685	36920
	VA	9745	4005	6205	5515	4255	4315	34040
	NDMS	20305	8360	12950	11530	8880	9015	71040
	Total	40610	16715	25880	23040	17740	18015	142000

Appendix F. Module Flow Diagrams

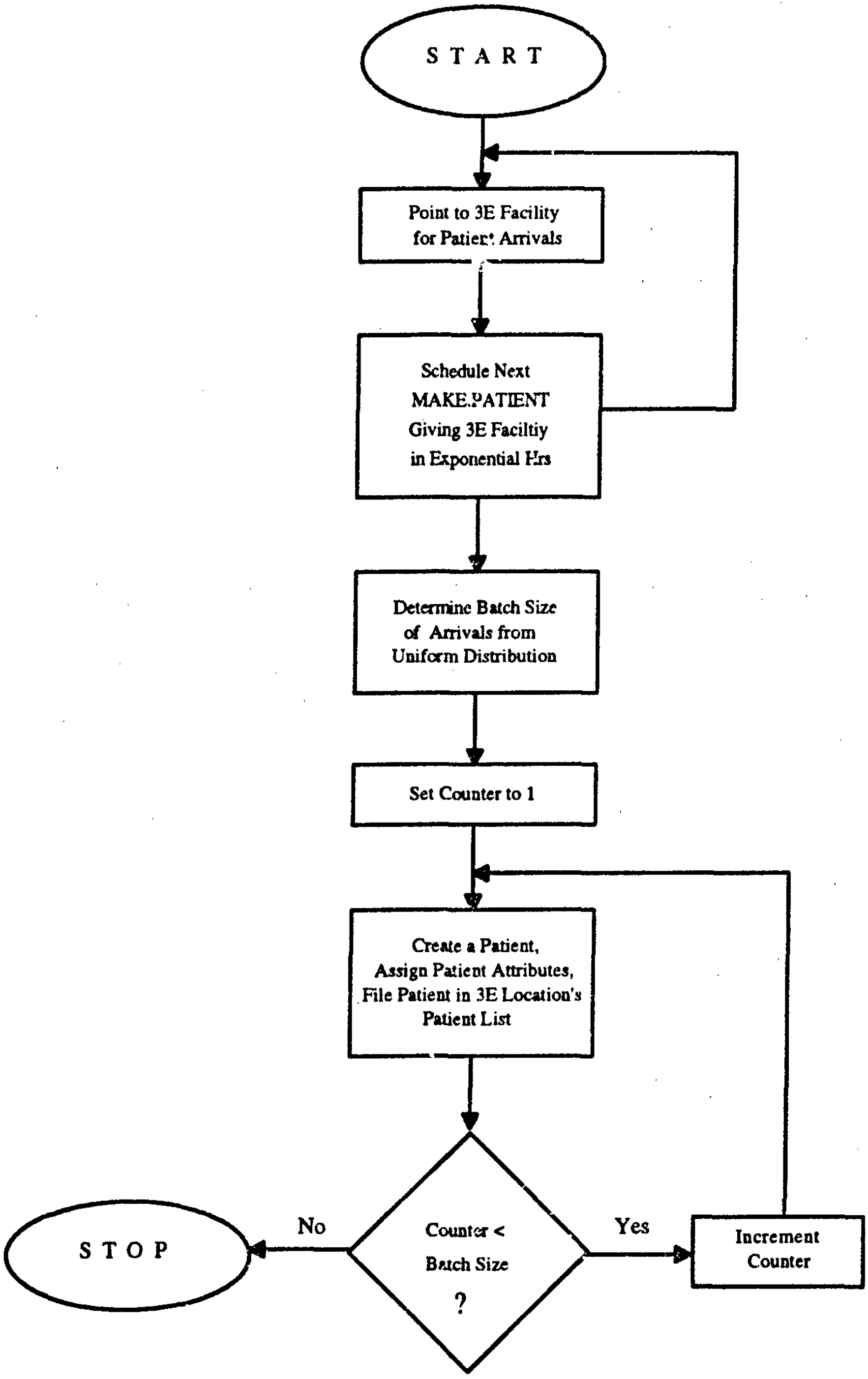
START
Point to 3E Facility for Patier.t Arrivals
Schedule Next MAKE.PATIENT Giving 3E Faciltiy in Exponential Hrs
Determine Batch Size of Arrivals from Uniform Distribution
Set Counter to 1
Create a Patient, Assign Patient Attributes, File Patient in 3E Location's Patient List
Counter < Batch Size ?
No
STOP
Yes
Increment Counter

Figure F.1. EVENT MAKE.PATIENT Flowchart

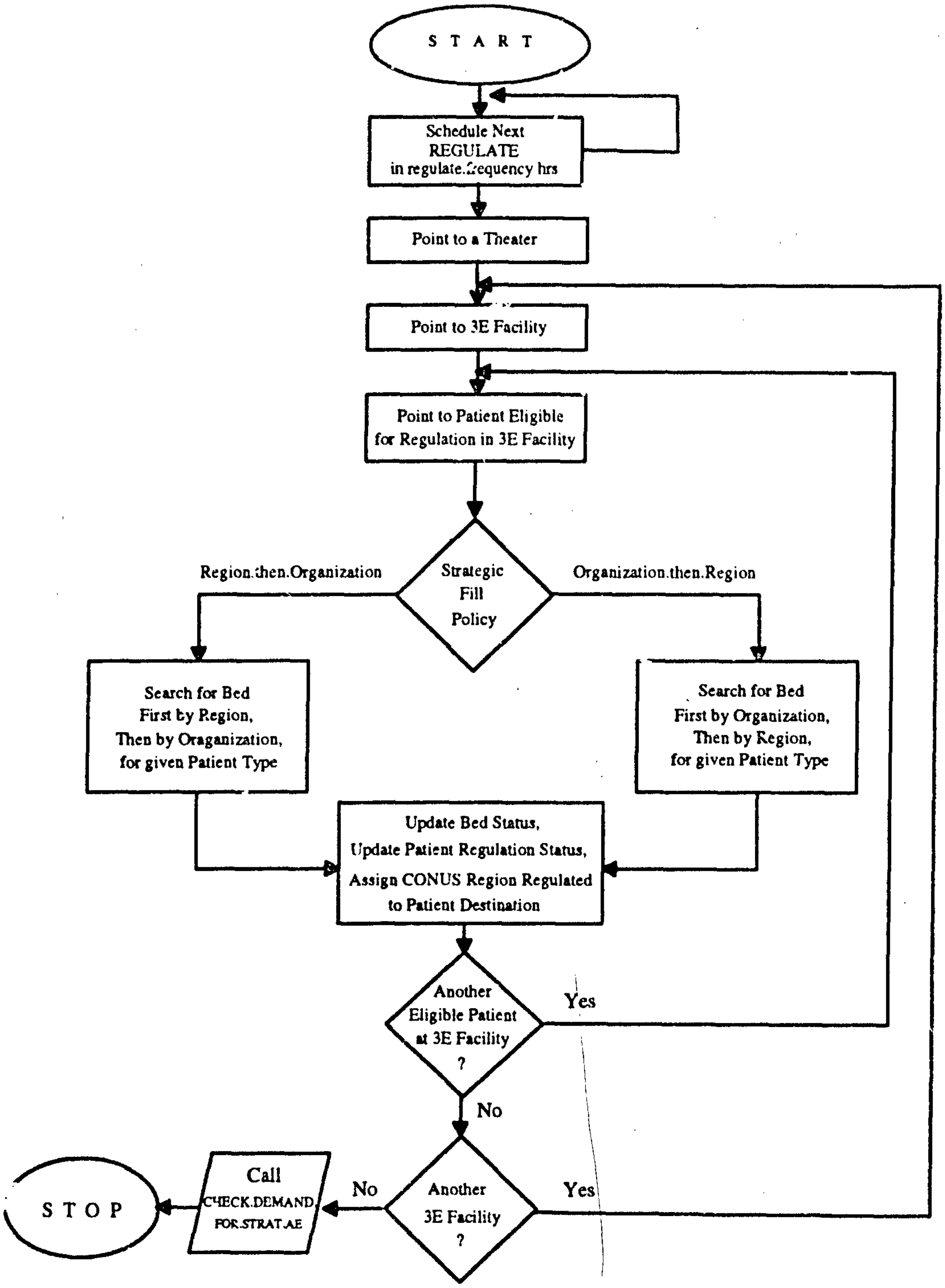
START
Schedule Next REGULATE in regulate.frequency hrs
Point to a Theater
Point to 3E Facility
Point to Patient Eligible for Regulation in 3E Facility
Strategic Fill Policy
Region.then.Organization
Organization.then.Region
Search for Bed First by Region, Then by Oraganization, for given Patient Type
Search for Bed First by Organization, Then by Region, for given Patient Type
Update Bed Status, Update Patient Regulation Status, Assign CONUS Region Regulated to Patient Destination
Another Eligible Patient at 3E Facility ?
Yes
No
Another 3E Facility ?
Yes
No
Call CHECK.DEMAND.FOR.STRAT.AE
STOP

Figure F.2. EVENT REGULATE Flowchart

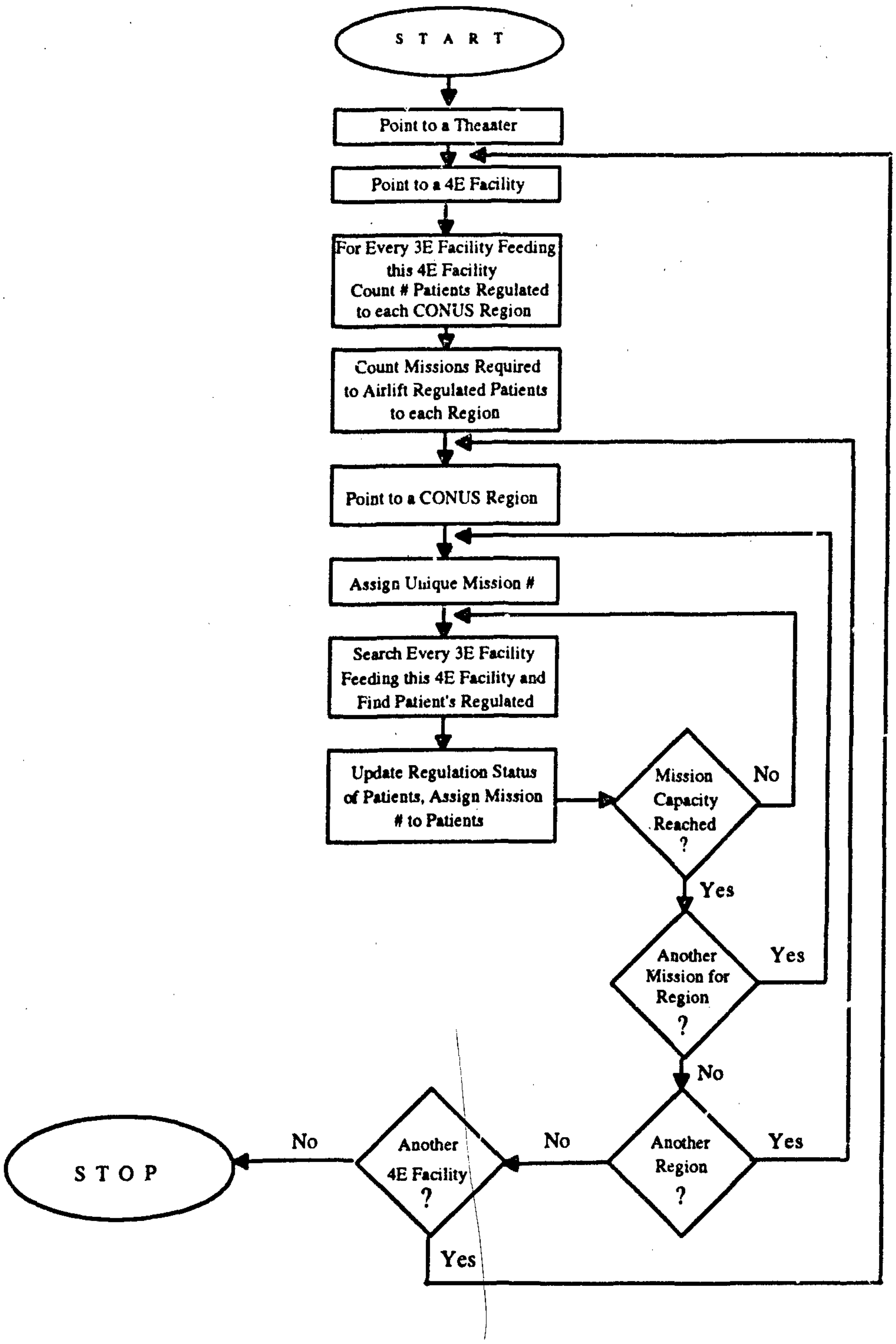
START
Point to a Theaater
Point to a 4E Facility
For Every 3E Facility Feeding this 4E Facility Count # Patients Regulated to each CONUS Region
Count Missions Required to Airlift Regulated Patients to each Region
Point to a CONUS Region
Assign Unique Mission #
Search Every 3E Facility Feeding this 4E Facility and Find Patient's Regulated
Update Regulation Status of Patients, Assign Mission # to Patients
Mission Capacity Reached ?
No
Yes
Another Mission for Region ?
Yes
No
Another Region ?
Yes
No
Another 4E Facility ?
No
Yes
STOP

Figure F.3. EVENT CHECK.DEMAND.FOR.STRAT.AE Flowchart

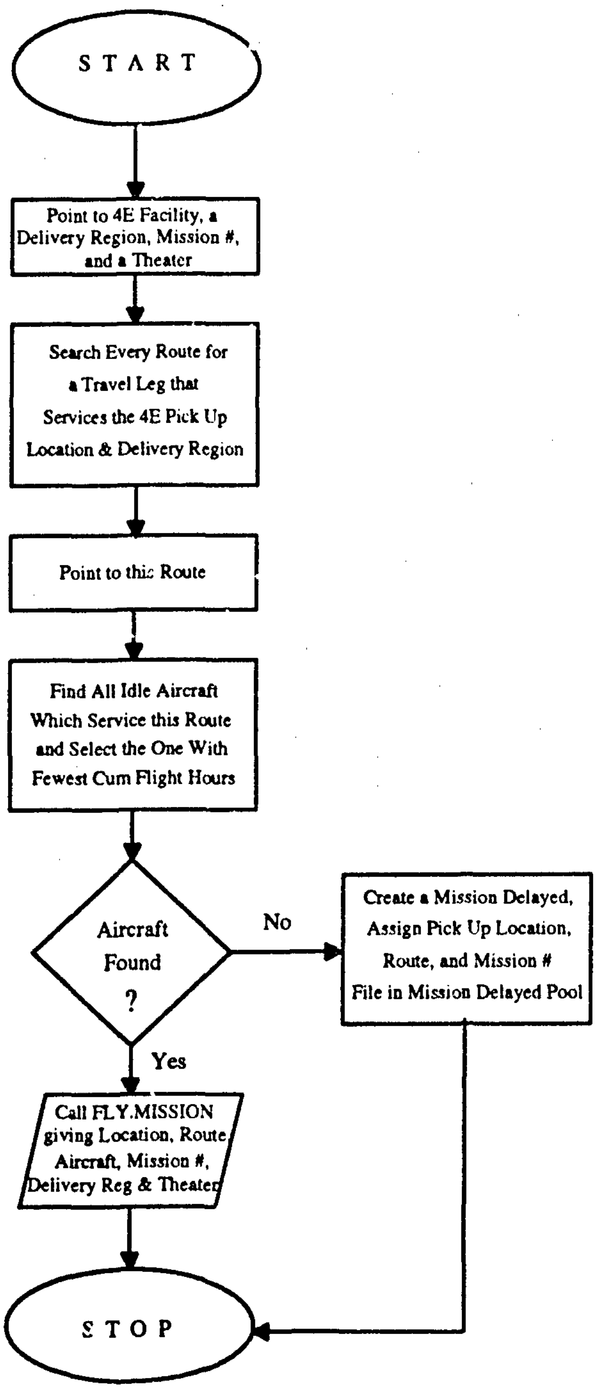
START
Point to 4E Facility, a Delivery Region, Mission #, and a Theater
Search Every Route for a Travel Leg that Services the 4E Pick Up Location & Delivery Region
Point to this Route
Find All Idle Aircraft Which Service this Route and Select the One With Fewest Cum Flight Hours
Aircraft Found ?
No
Create a Mission Delayed, Assign Pick Up Location, Route, and Mission # File in Mission Delayed Pool
Yes
Call FLY.MISSION giving Location, Route Aircraft, Mission #, Delivery Reg & Theater
STOP

Figure F.4. EVENT MISSION.GENERATOR Flowchart

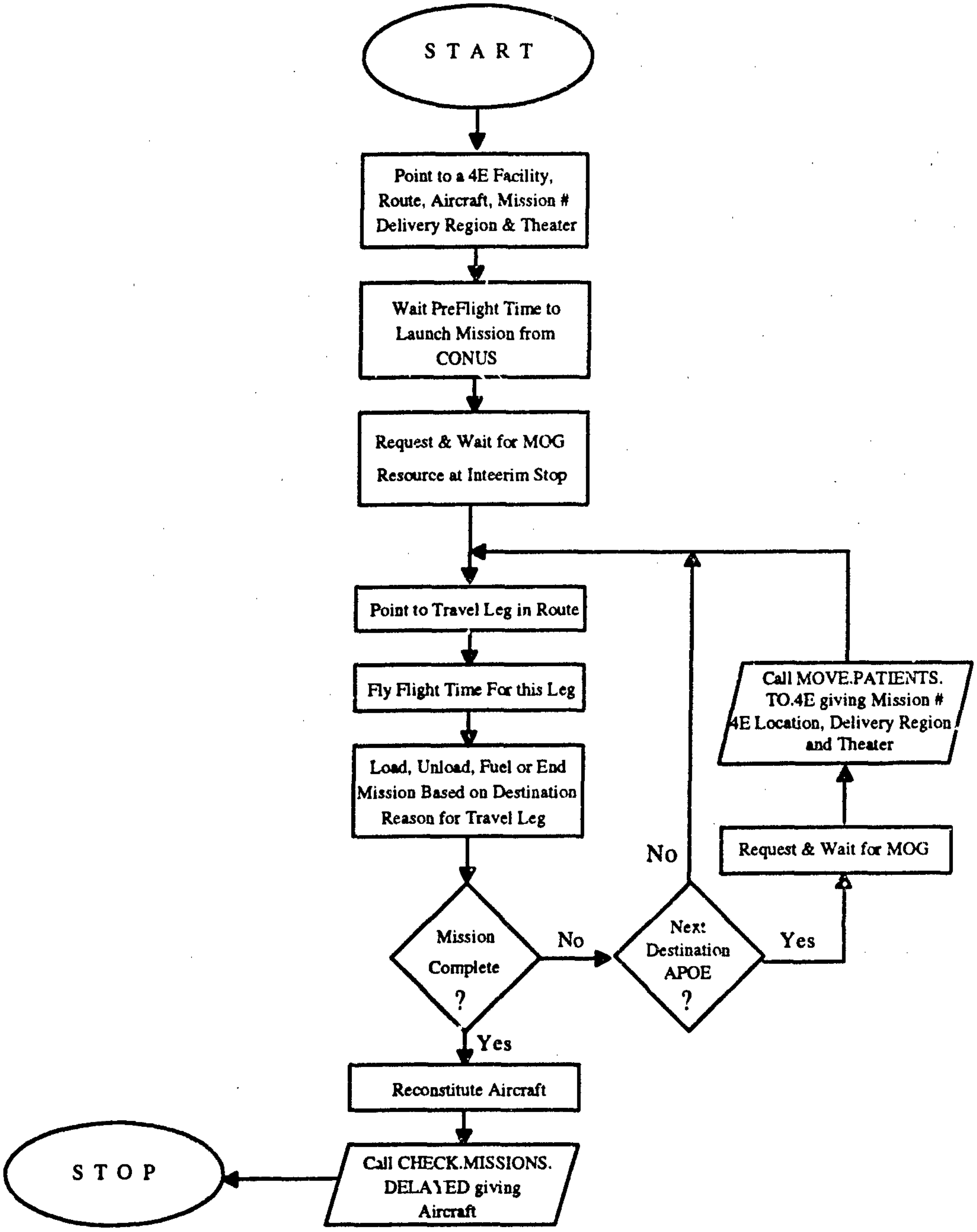
START
Point to a 4E Facility, Route, Aircraft, Mission # Delivery Region & Theater
Wait PreFlight Time to Launch Mission from CONUS
Request & Wait for MOG Resource at Inteerim Stop
Point to Travel Leg in Route
Fly Flight Time For this Leg
Load, Unload, Fuel or End Mission Based on Destination Reason for Travel Leg
Mission Complete ?
No
Next Destination APOE ?
Yes
Request & Wait for MOG
Call MOVE.PATIENTS. TO.4E giving Mission # 4E Location, Delivery Region and Theater
No
Yes
Reconstitute Aircraft
Call CHECK.MISSIONS. DELAYED giving Aircraft
STOP

Figure F.5. PROCESS FLY.MISSION Flowchart

Vita

Major Charles W. Wolfe, Jr. was born on 5 March 1958 in Ruston, Louisiana. He graduated from Woodward High School in Woodward, Oklahoma in 1976. After high school, he attended the United States Air Force Academy where he majored in Operations Research. He graduated with military distinction and was commissioned in 1980. He was immediately assigned as an Air-to-Air Missile Effectiveness Analyst at the Air Force Armament Laboratory, Eglin AFB, Florida. While there he also served as a project officer for the Boosted Kinetic Energy Penetrator Program and earned an M.B.A. from the University of West Florida. In August 1983, Major Wolfe was reassigned to the Air Force Operational Test & Evaluation Center in Albuquerque, New Mexico. There, as a Munitions Logistics Analysis Manager, he performed reliability, maintainability, and availability studies on several major USAF munitions programs. He was then assigned to Headquarters Air Force Systems Command and served the Directorate of Personnel first as a Career Development Program Analyst and later as Assistant for Information Systems Analysis. In January 1990, Major Wolfe joined the Commander's Staff Group where he served as Chief of Strategic Planning. Major Wolfe graduated from the Program Management Course at the Defense Systems Management College at Ft Belvoir, Virginia in June 1991. In August he entered the School of Engineering at the Air Force Institute of Technology. After graduation he will be assigned to the B-2 System Program Office at Wright-Patterson AFB, Ohio.

Major Wolfe married the former Geri Jean Converse of Woodward, Oklahoma in 1980. They have two children, Matthew Charles and Katheryn Ann.

Permanent address: 3708 Windover Drive
Norman, Oklahoma 73072

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
March 1993

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
THE USE OF SIMULATION TO EVALUATE AEROMEDICAL EVACUATION (AE) POLICY AND PLANNING

5. FUNDING NUMBERS

6. AUTHOR(S)
Charles W. Wolfe, Jr., Major, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENS/93M-26

9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ AMC/XPY
Scott AFB IL 62225

10. SPONSORING MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Strategic aeromedical evacuation (AE) of casualties from the theater of operations to the CONUS during wartime is a complex operation that involves the integration of medical personnel and policies with airlift concepts and capabilities. Military analysts within the Air Mobility Command Analysis Group (AMC/XPY) have traditionally used deterministic linear programming techniques to estimate the number of aircraft the United States Air Force (USAF) requires for given contingency scenarios. However, this group has yet to develop a stochastic approach to validate their resource recommendations, and more importantly, to study the interrelationships between key factors comprising strategic aeromedical evacuation. As the possibility for many smaller campaigns around the world increases, USAF medical planners require a flexible, analytical tool which captures the major elements of this important mission in order to quickly evaluate differing medical airlift plans and policies. This thesis develops, documents, and demonstrates the use of a computer simulation model for strategic AE operations that is modular in nature, completely data driven, and quickly adaptable to scenario changes, as a policy/planning aid for the AMC Surgeon and his staff. In addition, this thesis investigates the use of two statistical techniques, principal component analysis and factor analysis, for interpretation of the simulation output.

14. SUBJECT TERMS
Aeromedical Evacuation, Medical Airlift, Simulation, Multivariate Analysis, Factor Analysis

15. NUMBER OF PAGES
212

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL